

For Reference

NOT TO BE TAKEN FROM THIS ROOM

For Reference

NOT TO BE TAKEN FROM THIS ROOM

Ex LIBRIS
UNIVERSITATIS
ALBERTAENSIS





Digitized by the Internet Archive
in 2018 with funding from
University of Alberta Libraries

<https://archive.org/details/Kuhn1963>

THE UNIVERSITY OF ALBERTA

NONLINEAR SYSTEMS AND NONLINEAR PROGRAMMING

by

ARTHUR KUHN

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE
OF MASTER OF SCIENCE

DEPARTMENT OF MATHEMATICS

EDMONTON, ALBERTA

DATE... Aug. 15/63.

ABSTRACT

The advent of the digital computer has given a new impetus to the development of efficient mathematical methods and computational techniques of mathematical programming, i.e. the problem of optimizing a constrained multivariate function. Dantzig, in 1947, introduced the new discipline of Linear Programming and the Simplex method as a practical method of solving the linear programming problem. Linear Programming is concerned with the optimization of a linear function subject to linear constraints and a surprisingly large class of industrial and business problems could be formulated as such an optimization problem. A great interest remains, however, in the many problems which do not realistically fit into the framework of Linear Programming; this remaining field of nonlinear programming - the problem of optimizing a nonlinear function subject to linear or nonlinear constraints - is both of practical and theoretical interest.

Intimately associated with the mathematical programming problem is the problem of solving nonlinear systems - a problem relatively neglected in the literature. Included in this thesis along with a broad review of the classical and more recent mathematical literature in linear and nonlinear algebraic problems are the numerical and computational aspects of the algebraic and non-algebraic nonlinear systems. The methods considered include the extension of iterative techniques usually applied to linear systems and the more important "Gradient Methods" of optimization. The

more recent results which are critically analysed include the concepts of approximation of a nonlinear function in a region near a solution by a positive definite quadratic function; the exploitation of the special properties of such a function; and the conjugation of successive directed steps towards the solution.

Examination of the available literature revealed:

(i) that practical methods for solving nonlinear algebraic systems in several variables are not available,

(ii) that many methods are available for special nonlinear problems, viz quadratic functions,

(iii) that many of these methods have been inadequately tested or are not applicable to a wide range of problems,

(iv) that algorithms for digital computers have only rarely been given, and

(v) that but very few numerical methods can be made readily accessible to the practising numerical analyst.

In this thesis the concentration is on (iii), (iv) and (v) i.e. on testing, devising algorithms, and the more specific presentation of the available mathematical literature applicable to the nonlinear systems and nonlinear programming problems.

ACKNOWLEDGEMENTS

I wish to thank Professor J. McNamee for his unfailing interest in this thesis. The broad scope of the topics covered and the introduction of computing algorithm notation at his suggestion have contributed greatly to the balance of the thesis. I would also like to acknowledge the consideration shown me by the Director and staff of the Computing Centre of the University of Alberta.

TABLE OF CONTENTS

		Page
CHAPTER I	INTRODUCTION	1
§ 1.1	Historical Background	1
§ 1.2	Mathematical Programming	5
§ 1.3	Survey of the Thesis	6
CHAPTER II	CLASSICAL ANALYTIC AND ITERATIVE METHODS FOR SOLVING LINEAR AND NONLINEAR ALGEBRAIC PROBLEMS	10
§ 2.1	Introduction	10
§ 2.2	Definitions and Notation	11
§ 2.3	Polynomials in One Variable	14
§ 2.31	Theorems, Lemmas, and Algorithms	14
§ 2.32	Upper Bounds to the Real Roots of a Polynomial	24
§ 2.33	Approximate Determination of Roots	26
§ 2.33.01	The Graphical Method	26
§ 2.33.02	Graeffe's Root-squaring Process	26
§ 2.33.03	Aitken-Bernoulli Method	28
§ 2.33.04	Whittaker's Theorem	30
§ 2.34	Improving the Accuracy of Roots	30
§ 2.34.01	Method of False Position	31
§ 2.34.02	Newton-Raphson Method	31
§ 2.34.03	Cubically Convergent Formulae	32
§ 2.34.04	Bairstow's Method	32
§ 2.35	Evaluation of zeros of Ill- conditioned Polynomials	33

§ 2.4	Polynomials in Several Variables	35
§ 2.41	Theorems, Definitions, and Algorithms	35
§ 2.42	Sylvester's Dialytic Method of Elimination	37
§ 2.5	Curve Tracing of Polynomials in Two Variables	39
§ 2.51	De Gua Triangle	39
§ 2.52	Properties of the Analytic Triangle	40
§ 2.53	Examples of the Method	42
§ 2.6	Systems of Linear Equations	43
§ 2.61	Theorems, Definitions, and Notation	44
§ 2.62	Definition of the Problem of Linear Algebra	48
§ 2.63	Direct Methods	49
§ 2.63.01	Cramer's Rule	49
§ 2.63.02	Gauss Method	49
§ 2.64	Iterative Methods	51
§ 2.64.01	Simple Iteration	51
§ 2.64.02	Gauss Seidel Iteration	52
§ 2.64.03	Example I	52
§ 2.65	Gradient and Conjugate Gradient Iterative Methods	53
§ 2.65.01	The Gradient Method	54
§ 2.65.02	Conjugate Gradient Method	55

CHAPTER III	TECHNIQUES FOR SOLVING NONLINEAR	
	SYSTEMS	57
§ 3.1	Introduction	57
§ 3.2	Definitions and Theorems	59
§ 3.21	Definitions and Notation	59
§ 3.22	Theorems and Lemmas	61
§ 3.3	Simple and Gauss-Seidel Iteration	61
§ 3.31	Simple Iteration	62
§ 3.32	Example of Simple Iteration	62
§ 3.33	Gauss-Seidel Iteration	63
§ 3.4	Gradient Methods	63
§ 3.41	Univariate or Relaxation Methods	64
§ 3.42	Newton's Method	68
§ 3.43	Methods of Descent	69
§ 3.43.01	Method of Steepest Descent	70
§ 3.43.02	Example of Steepest Descent	72
§ 3.44	Steepest Descent by Integration of a Set of Ordinary Differential Equations	74
§ 3.45	Conjugate Gradient Method of Steepest Descent	75
§ 3.46	A Mixed Method	86
§ 3.5	Direct Search	88
§ 3.51	Downhill Method by Ward and Lance	88
CHAPTER IV	LINEAR PROGRAMMING	92
§ 4.1	Introduction	92
§ 4.2	Definition of the Problem	94

§ 4.3	Definitions and Theorems	94
§ 4.31	Definitions	94
§ 4.32	Theorems	94
§ 4.4	The Simplex Method	99
§ 4.41	Formulation of the Simplex Method	99
§ 4.42	The Dual Simplex Method	103
§ 4.5	The Transportation Problem	104
CHAPTER V	NONLINEAR PROGRAMMING	107
§ 5.1	Introduction and Definition of the Problem	107
§ 5.11	Introduction	107
§ 5.12	Definition of the Problem	108
§ 5.2	Definition and Theorems	108
§ 5.21	Definitions and Notation	108
§ 5.22	Theorems and Lemmas	110
§ 5.3	Quadratic Programming	111
§ 5.31	Convex Programming by Extension of Simplex Method	111
§ 5.31.01	Example	115
§ 5.31.02	The Basic Computing Algorithm	120
§ 5.32	Method of Feasible Directions	122
§ 5.32.01	Explicit Formulation of Method of Feasible Directions	124
§ 5.32.02	Determination of Optimum Feasible Directions	127
§ 5.33	Gradient Projection Method	130
§ 5.33.01	Preview of Calculations	131

§ 5.33.02	The Basic Computing Algorithm	140
§ 5.33.03	Example of Gradient Projection	
	Method	141
§ 5.33.04	Further Considerations of the	
	Method	145
CHAPTER VI	RESULTS AND CONCLUSIONS	152
§ 6.1	The Approach Used in the Thesis	152
§ 6.2	Review of Thesis and Conclusions	153
§ 6.3	Remaining Problems	155
BIBLIOGRAPHY		157

LIST OF TABLES

	Page
TABLE I	
Comparison of Results Between Conjugate Gradient and Other Methods	79
TABLE II	
An Example of an Iterative Cycle for Conjugate Gradient Method	80

NOTATION

Throughout this thesis use is made of matrix notation. The dimensions of a matrix or vector used in any section are shown as soon as possible after their first appearance in that section. A matrix or vector may be modified by a superscript or subscript which may have a special significance in its use.

Matrix Notation

$A : nxm$	a matrix composed of n rows and m columns.
$x : nx1$	a set of n variables x_1, x_2, \dots, x_n which define a point or n -dimensional column vector in n space.
$x^T : 1xn$	a row vector obtained by transposition of a column vector.
$N_q : nxq$	the subscript here gives added information on the size of the matrix.
$I : nxn$	a special matrix frequently used defined as the identity matrix $[\delta_{ij}]$, where $\delta_{ij} = 1$, if, and only if, $i = j$.

The following notation is used in the description of the computing algorithms and has a special significance in computer applications where the assumption is made that any quantity, e.g. a matrix, vector, number, etc. may be referred to by a symbol.

Algorithmic Notation

$\alpha \leftarrow \beta$	the quantity α is set equal to β
$P ; \alpha \leftarrow \beta$	the notation $P ;$ labels the associated instruction as P .
$\rightarrow P$	the next instruction is the one labelled P .

$\alpha : \beta , \xrightarrow{r} P$ the next instruction is P if α r β is satisfied where r is a relation such as $< , \neq ,$ etc.

$N \oplus i$ the quantity i is concatenated (designated by the \oplus symbol) with N ; e.g. if $N : m \times n , i : m \times 1$ then $N \oplus i : (m+1) \times n$.

$g \Longleftrightarrow \nabla f(x)$ the equivalence symbol \Longleftrightarrow is used to condense material or imply a whole series of operations.

CHAPTER I

INTRODUCTION

The advent of the large high-speed electronic computer and the desire to optimize systems have given a new impetus to the study of the optimization problem. Interest in this problem arose in economics, transportation, and management sciences. One aim of this thesis is to review and analyse critically the classical and more recent contributions to the problems of solving systems of nonlinear equations and optimizing a nonlinear function subject to constraints. A second aim is to use this work in developing practical methods for high-speed computers.

The introductory chapter sketches in the historical background to the solution of equations and systems of equations; and reviews the growth of mathematical programming: finally, the contents of each chapter are summarized.

§ 1.1 Historical Background

At the outset we shall attempt to place the nonlinear problem in its proper historical context. The Egyptian, Greek, Arabian, and later the medieval mathematicians were restricted to rational operations and the extraction of roots; they were able to solve the general polynomial of degree $n \leq 4$. The first rigorous proof that solution by radicals would be insufficient for polynomials of degree $n \geq 5$ was published in 1824 by Abel. This indicates that an important line of division lies between polynomial equations of the first four degrees and those of higher degree. The unification of algebra and geometry, i.e. analytic geometry, can be attributed to Descartes "Géométrie" in 1637. A deeper insight into the nature and properties of an algebraic equation was given by the theory of substitutions discovered by Lagrange in 1770 and finally by

the theory of groups attributed to Galois, Cauchy, et. al. since 1844-46. Although the fundamental theorem of algebra was posed at least two centuries earlier, a rigorous proof was not available until Gauss in 1799.

It is known that a special case of a system of linear equations had been solved by the early Greeks but a satisfactory treatment of linear systems was not possible until determinants had been developed. This is a comparatively recent topic in algebra, having its origin in the writings of Leibnitz at the end of the seventeenth century, and assuming a significant position in mathematical literature during the latter part of the eighteenth and first part of the nineteenth century. Sylvester's dialytic method of elimination is a special application of the theory of determinants to the nonlinear algebraic system. This method converts the nonlinear system into a problem of a polynomial equation in one variable. The dialytic method is of limited practical value and is restricted to systems of low order and low degree; e.g., the second order algebraic system $f_n(x,y) = 0$, $f_m(x,y) = 0$, of total degree n , m , respectively, has a resultant polynomial of degree $\leq mn$. Another important application of the theory of determinants is the solution of a nonlinear system by iterative techniques.

The numerical solution of algebraic equations remained a very difficult problem even after the explicit expression for the roots of the polynomial was known. This computational difficulty led to a search for practical and particular

methods as opposed to general methods. For example, Cardan's solution of the cubic requires the evaluation of the following expression:

$$\left(\frac{-g+k\sqrt{-1}}{2}\right)^{\frac{1}{3}} + \left(\frac{-g-k\sqrt{-1}}{2}\right)^{\frac{1}{3}}.$$

Now there is no general arithmetical process for extracting the cube root of such complex numbers, and consequently this formula is useless for purposes of arithmetical calculation. On an electronic computer the solving of the cubic equations may be handled by the unsophisticated but simple method of bisections. This serves as a good example of the distinction between the analytic and practical viewpoint.

We may observe that for many applications only approximate values of the real roots are needed. We will assume for the sake of brevity that we need consider only equations with real coefficients. Vieta's method of obtaining the approximate real roots of a polynomial was published in 1600 and finally perfected as Horner's method in 1821. Sturm's theorem (proved in 1829) furnishes the scientific foundation for every method of finding the approximate values of the roots in an algebraic equation with real coefficients; it determines the number of real roots between two arbitrarily assigned numbers. In 1767, Lagrange published a theoretically simple method for finding the approximate value of an irrational root by means of continued fractions where the required information is knowledge of the existence of a real root within an interval. Although this method is perspicuous and exhibits

clearly the reason for each step, it has not been used as widely as the well-known Horner's method.

A useful technique in solving certain algebraic problems is by graphic methods. These methods were developed mostly since the beginning of the nineteenth century and they serve either as a rough check on the accuracy of calculations or as a sufficiently accurate solution for the problem in hand. The literature on graphic algebra is very extensive and among the introductory treatises we may mention the Graphic Algebra by Phillips and Beebe.

The solution of transcendental functions poses its own special problems since we have to consider the possibility of an infinite number of roots or no roots whatsoever. For example, the function $\cos x$ has an infinite number of real roots ($x = \pm k\pi + \frac{\pi}{2}$, $k=0,1, \dots$), but e^x has neither real nor complex roots. Some of the methods available for finding the roots of polynomials can be extended to solve the transcendental problems.

The problem of solving systems of nonlinear non-algebraic equations has been relatively neglected in the mathematical literature. The method of iteration can quite often be successfully applied to the solution of the nonlinear system. A recent approach to this problem is to solve for the minimum of an associated test function. The methods usually employed are known as gradient methods. The solution of the

nonlinear system is much more complicated than the solution of one equation in one unknown. The basic difficulty is the lack of effective methods for the finding of initial approximate solutions. Newton's method, Simple, and Gauss - Seidel are usually effective for making solutions more accurate.

§ 1.2 Mathematical Programming

Mathematical programming is concerned with the problem of maximizing or minimizing a function of several variables that are restricted by a number of constraints. The problem of optimizing such a function restricted to equality constraints is intimately associated with the general optimization problem. The classical technique applied to the equality constrained optimization problem is Lagrange's method of Undetermined Multipliers. Although some success has been achieved in adapting Lagrange's method to modern programming problems, the trend in the last few years has been to evolve gradient and conjugate gradient methods as the most efficient methods of computational attack.

The theoretical and computing aspect of the optimization problem subject to constraints has received renewed attention in the last two decades and given rise to the new disciplines of linear and nonlinear programming. In 1947, Dantzig in cooperation with Wood was concerned with an analysis of military programming and he proposed that the

interrelationships between the activities of a large organization should be viewed as a model of Linear Programming in which the best program is determined by minimizing a linear function subject to linear constraints. That this concept of linear programming could be used in other contexts was soon recognized and many mathematicians became interested in the theoretical and computational aspects of this field. The fundamental literature in this field may be found in Koopmans (1951) which includes the work of Dantzig (1951 a, b)- the originator of the basic principles of Linear Programming. In many problems a far more realistic formulation is possible if a nonlinear model is used. In such cases the optimum solution obtained by a nonlinear programming method should have considerably more validity than the solution to the approximate linear programming problem. Examples of two possible new areas of application are optimum design of process equipment, and the direct solution of problems in mechanics, physics and chemistry which require that the energy (given as a function of the system variables) be minimized subject to one or more subsidiary conditions, Rosen (1960). The most recent work in the field of nonlinear programming is reviewed in Spang (1962), Turner (1960), Riley and Gass (1958), Wolfe (1962), and Zoutendijk (1960).

§ 1.3 Survey of the Thesis

The subject matter of this thesis is shown in detail in the "table of contents" but, nevertheless, it will not be

superfluous to make a short survey of each chapter. The first chapter consists of the introduction.

Chapter II reviews the more pertinent classical theorems, definitions, and techniques concerned with linear and nonlinear algebraic equations and systems of equations. Numerical solutions and iterative techniques which may be applied to these algebraic problems are also examined. Graphical methods - which had a considerable vogue before digital computers became available and are still sometimes useful in analogue computers - are considered as are problems concerned with curve tracing. In preparation for the more difficult nonlinear problems, we review the classical problems of solving the linear system of n equations in n unknowns. The particular methods reviewed include the Gauss Elimination technique, Simple iteration, and Gauss-Seidel iteration.

In Chapter III we review the problem of solving systems of nonlinear algebraic and non-algebraic equations. The question of existence of solutions for the algebraic system is satisfactorily answered by converting the nonlinear algebraic system into a polynomial in a single variable by the dialytic method of elimination. The computational considerations of the dialytic method suggest that this method is in general unsuitable as the basis for a numerical solution. Numerical methods which may be effectively applied to the nonlinear problem include the extensions of Simple and Gauss-Seidel iteration methods and the use of gradient or

optimization techniques. A more recent approach is the almost experimental method of direct search.

The basic concepts and elementary techniques of Linear Programming are outlined in Chapter IV. This review attempts to include the more fundamental theorems and methods in this new discipline. We review the basic Simplex and Dual Simplex methods but do not attempt to extend or evaluate the other methods currently available.

In Chapter V we review the more difficult mathematical programming problem - nonlinear programming. A nonlinear programming problem is obtained when the linearity restriction is removed from the objective function and constraints. We examine in some detail a special case of nonlinear programming defined as quadratic programming - the problem of minimizing a convex quadratic function (or its dual), subject to linear constraints - in great detail emphasizing the importance of gradient methods.

In the last Chapter we attempt to summarize the conclusions applicable to the nonlinear problem. A comparison of methods on any particular problem or group of problems is not very helpful since there will always be some particular function for which a given method is best suited. We are therefore left with the exercise of experience and intuition in order to fit the appropriate method to the particular problem.

The organization of the succeeding Chapters is as follows: an introduction; definitions, notation, and theorems; methods and examples where applicable. The computing algorithm - a device which describes the numerical calculational procedure - is freely used to challenge the interest of the reader and yet present the numerical calculation in a simple and precise manner.

CHAPTER II
CLASSICAL ANALYTIC AND ITERATIVE METHODS FOR SOLVING
LINEAR AND NONLINEAR ALGEBRAIC PROBLEMS

§ 2.1 Introduction

The roots of an arbitrary polynomial in one variable $P_n(x)=0$ can be given in explicit form for $n \leq 4$. For $n \geq 5$ the explicit formulae are in general expressed in terms of hyperelliptic functions; they are often complicated and unwieldy and are rarely used as the basis of a numerical method of solution. Classical methods are available for the numerical solutions of the linear algebraic problem $Ax=b$ ($A:n \times n$ matrix; $x, b:n \times 1$ vectors) and the polynomial equation in one variable $P_n(x)=0$; and some results are available for the more complicated set of nonlinear algebraic equations in n variables $F(x)=0$ ($F(x), x:n \times 1$ vectors). Questions of uniqueness and existence of solutions for these algebraic problems have been extensively considered in the literature which is scattered and incomplete.

In this chapter the more fundamental and well-known together with some less-known but useful results in this field are reviewed and their pertinence to this thesis is briefly discussed in the succeeding chapters.

The important questions arising from ill-conditioning (i.e. polynomials whose zeros are oversensitive to small

changes in the coefficients) is too large a topic to be included within the scope of this thesis. Wilkinson (1959) has shown that very useful results can be obtained for the polynomial case by quite simple analysis. We discuss his work briefly but do not attempt to discuss the extensions of his work to the very difficult problems of ill-conditioning for nonlinear algebraic problems in many unknowns.

Once again, it may be emphasized that the classical techniques considered in this chapter are not always good techniques for electronic computers.

The principle of iteration was used as early as 1674 for the determination of square roots. This principle has since then been extended to finding the roots of polynomials, solving systems of equations, etc., and is becoming increasingly important as an approach for the numerical solution of algebraic and transcendental equations. We study some of the results in this field.

§ 2.2 Definitions and Notation

§ 2.2.01 We shall use $P_n(x)$ to denote a polynomial in x of degree n expressed in the normal form

$$P_n(x) = a_0 x^n + a_1 x^{n-1} + \dots + a_{n-1} x + a_n, \quad a_0 \neq 0, \quad (1)$$

where a_0 is the coefficient of x^n and the polynomial is ordered in descending powers of x .

§ 2.2.02 The previous notation can be readily generalized to a polynomial in n variables x_1, x_2, \dots, x_n and expressed in the normal form

$$f(x_1, x_2, \dots, x_n) = g_0(x_2, \dots, x_n)x_1^m + \dots + g_{m-1}(x_2, \dots, x_n)x_1 + g_m(x_2, \dots, x_n) \quad (2)$$

where m is the highest degree of any term containing x_1 and the $g(x)$'s are polynomials in the n-1 variables (x_2, \dots, x_n) . We may interchange the role of $x_1, i \neq 1$, with x_i in (2) and obtain a similar expression in x_i . The highest degree in x_i of any term in the polynomial in the form (2) whose coefficient is not zero is called the degree of the polynomial in x_i , and the highest total degree of any term whose coefficient is not zero is called the degree of the polynomial. The particular polynomial equation in n variables where m is the maximum sum of the exponents of any one term, i.e. the polynomial of total degree m, is denoted by

$$f_m(x_1, x_2, \dots, x_n) = 0.$$

§ 2.2.03 The n equations of the first degree in n unknowns

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + \dots + a_{2n}x_n &= b_2 \\ \dots & \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n \end{aligned} \quad (3)$$

where a's are any constant, real or complex, is defined as a system of n linear simultaneous algebraic equations in n unknowns. The system of equations (3) may be denoted more

briefly by

$$a_{ij}x_j = b_i \quad i=1, \dots, n; j=1, \dots, n,$$

or in matrix form as

$Ax = b$, where A is an $n \times n$ matrix; b, x ; $n \times 1$ column vectors.

§ 2.2.04 We can extend the notation for linear systems to nonlinear systems as

$$f_1(x_1, x_2, \dots, x_n) = b_1$$

$$f_2(x_1, x_2, \dots, x_n) = b_2$$

.....

$$f_n(x_1, x_2, \dots, x_n) = b_n \quad \text{or as}$$

$F(x) = b$, where $F(x)$, x , and b , are $n \times 1$ column vectors.

§ 2.2.05 A Sturmian sequence is a sequence of $n+1$ functions $f_0(x), f_1(x), \dots, f_n(x)$ which satisfy on a given interval $[a, b]$ of the real axis the following conditions:

1. $f_i(x)$ are continuous functions ($i=0, 1, \dots, n$).
2. $\text{Sign } f_n(x) = \text{constant}$ for $x \in [a, b]$.
3. If $f_i(x) = 0, f_{i+1}(x)$ and $f_{i-1}(x) \neq 0$ for $x \in [a, b]$ and all i .
4. If $f_i(x) = 0$, $\text{sign } f_{i-1}(x) = \text{sign } f_{i+1}(x)$ ($i=1, \dots, n-1$).
5. If $x_1 = x$ is a root of $f_0(x)$, then for h sufficiently small,
$$\frac{\text{sign } f_0(x-h)}{f_1(x-h)} = -1, \quad \frac{\text{sign } f_0(x+h)}{f_1(x+h)} = 1, \quad \text{Todd (1962).}$$

§ 2.3 Polynomials in One Variable

The following review of the classical results for polynomials in one variable is certainly desirable before we consider the more difficult nonlinear algebraic problem. A more extensive treatment may be found in Burnside and Panton (1912), Bocher (1922), and Turnbull (1944).

§ 2.31 Theorems, Lemmas, and Algorithms

§ 2.31.01 Fundamental Theorem of Algebra

Every polynomial equation $P_n(x)$ of degree n ($n \geq 1$) has at least one root.

Corollary:

Every polynomial equation $P_n(x)$ of degree n ($n \geq 1$) has exactly n roots.

§ 2.31.02 Factor Theorem

If x_1 is a root of $P_n(x)=0$ then $x-x_1$ is a factor of $P_n(x)$; i.e. $P_n(x) = (x-x_1) P_{n-1}(x)$

§ 2.31.03 Remainder Theorem

If a polynomial $P_n(x)$ is divided by the linear function $(x-x_1)$, where x_1 is any constant, the remainder R is defined by $R = P_n(x_1)$.

§ 2.31.04 Descartes' Rule of Signs

The polynomial equation $P_n(x)=0$ cannot have more positive real roots than there are variations in sign of successive coefficients of $P_n(x)$ or more negative roots than there are variations in sign of $P_n(-x)$.

§ 2.31.05 Theorem

If the polynomial $P_n(x)$ is evaluated at $x=a$ and $x=b$, where $a < b$ and both a and b real, and if $\text{sign } P_n(a) \neq \text{sign } P_n(b)$ then there exists at least one $x=x_0$ such that $P_n(x_0)=0$ where $a < x_0 < b$.

§ 2.31.06 Sturm's Theorem

Although the Descartes Rule of Signs is remarkable, an uncertainty as to the exact number of real roots in an equation still remains. The problem of finding a definitive test was finally solved in 1829 by Sturm. Sturm showed how to find for any polynomial equation the exact number of real roots which lie within any given range of values using only rational operations.

Theorem: There exists a set of real polynomials $P_n(x)$, $P_n'(x)$, $f_2(x), \dots, f_n(x)$ whose degrees are in descending order, such that if $b > a$, the number of distinct real roots of $P_n(x)=0$ between $x=a$ and $x=b$ is equal to the excess of the number of changes of sign in the sequence $P_n(x)$, $P_n'(x)$, $f_2(x), \dots, f_n(x)$ when $x=a$ over the number of changes of sign when $x=b$.

Corollary: If $f_r(x)$ is a remainder involving x and such that $f_r(x)$ remains positive, or remains negative, for all real values of x between a and b , then the sequence need not be prolonged beyond $f_r(x)$.

§ 2.31.07 Examples of Sturm's Theorem

Example 1: Find the number and situation of the real roots of the polynomial equation

$$P_3(x) = x^3 - 7x + 7 = 0.$$

Now $P_3'(x) = 3x^2 - 7,$

$$f_2(x) = P_3(x) - q_1 P_3'(x) = 2x - 3,$$

$$f_3(x) = P_3'(x) - q_2 f_2(x) = 1.$$

Hence we construct the following table:

x	$P_3(x)$	$P_3'(x)$	$f_2(x)$	$f_3(x)$
$-\infty$	-	+	-	+
-4	-	+	-	+
-3	+	+	-	+
+1	+	-	-	+
+2	+	+	+	+
∞	+	+	+	+

By inspection, we see that one real root lies in $[-4, -3]$ and two real roots are in $[1, 2]$.

Example 2: Find the nature of the roots of

$$P_4(x) = x^4 - 5x^3 + 9x^2 - 7x + 2 = 0.$$

Now $P_4'(x) = 4x^3 - 15x^2 + 18x - 7,$

$$f_2(x) = P_4(x) - q_1 P_4'(x) = x^2 - 2x + 1 = (x-1)^2,$$

$$f_3(x) = P_4'(x) - q_2 f_2(x) = 0.$$

Hence

x	$P_4(x)$	$P_4'(x)$	$f_2(x)$
$-\infty$	+	-	+
$+\infty$	+	+	+

We observe that the given equation has only two distinct real roots; but one of these is a triple root, as is evident from the form of $f_2(x) = (x-1)^2$.

§ 2.31.08 Euclid's Algorithm

From the Remainder Theorem we have

$$P_n(x) = P_{n-1}(x) (x-h) + R, \quad R = P_n(h).$$

A very useful and powerful method of describing a numerical computation is by the computing algorithm; this technique is used extensively throughout this thesis. We choose a compact symbolic notation which is almost self-explanatory. The use of this notation - outlined in the notation table - enables us to circumvent long periphrastic explanations. The computing algorithm using the symbolic notation is now used to describe the well-known Euclid's Algorithm for determining the remainder when a polynomial is divided by a linear factor.

Let n, h, a : $(n+1)x1$ define the problem; i.e.

$$P_n(x) = a_0x^n + a_1x^{n-1} + \dots + a_n.$$

The remainder $R = P_n(h)$ is then given by:

$$k \leftarrow 0$$

$$1; k \leftarrow k + 1$$

$$a_k \leftarrow a_k + a_{k-1}xh$$

$$n:k \quad \xrightarrow{\geq 1}$$

$$R \leftarrow a_n$$

END.

e.g. $P_4(x) = 3x^4 - 5x^3 + 10x^2 + 11x - 61; h = 3, n = 4.$

$$k \leftarrow 0$$

$$k \leftarrow 1$$

$$a_1 \leftarrow -5 + 3x^3 = 4$$

$$1 < 4$$

$$k \leftarrow 2$$

$$a_2 \leftarrow 10 + 4x^3 = 22$$

$$2 < 4$$

$$k \leftarrow 3$$

$$a_3 \leftarrow 11 + 22x^3 = 77$$

$$3 < 4$$

$$k \leftarrow 4$$

$$a_4 \leftarrow -61 + 77x^3 = 170$$

$$4 \nless 4$$

$$R \leftarrow a_4 = 170$$

END.

§ 2.31.09 Algorithm for Increasing a Root by h

This problem is essentially equivalent to Horner's method of finding a numerical solution of a polynomial. Hence, given $n, h; a: (n+1) \times 1$, the resultant polynomial is given by the following:

<u>conditional</u>		
	<u>branch</u>	<u>comment</u>
<hr/>		
$\ell \leftarrow n$		
1; $k \leftarrow 0$		
2; $k \leftarrow k+1$		
$a_k \leftarrow a_k + a_{k-1} x^h$		
$\ell : k$	$, \geq 2$	is stage 1 complete ?
$\ell \leftarrow \ell - 1$		
$\ell : 1$	$, \geq 1$	is stage 2 complete ?
END		
<hr/>		

e.g. $x^5 - 3x^4 - 24x^3 + 95x^2 - 46x - 101; h = 1, n = 5$

$$\ell \leftarrow 5$$

$$k \leftarrow 0$$

$$k \leftarrow 1$$

$$a_1 \leftarrow -3 + 1x1 = -2$$

$$1 < 5$$

$$k \leftarrow 2$$

$$a_2 \leftarrow -24 + -2x1 = -26$$

$$2 < 5$$

$$k \leftarrow 3$$

$$a_3 \leftarrow -26 + 95x1 = 69$$

$$3 < 5$$

$$k \leftarrow 4$$

$$a_4 \leftarrow 69 + -46x1 = 23$$

$$4 < 5$$

$$k \leftarrow 5$$

$$a_5 \leftarrow 23 + -101x1 = \underline{-78}$$

5 = 5 , stage 1 is complete

$$\ell \leftarrow 4$$

$$1 < 4$$

$$k \leftarrow 0$$

$$k \leftarrow 1$$

$$a_1 \leftarrow -2 + 1x1 = -1$$

$$1 < 4$$

$$k \leftarrow 2$$

$$a_2 \leftarrow -26 + -1x1 = -27$$

$$2 < 4$$

$$k \leftarrow 3$$

$$a_3 \leftarrow 69 + -27x1 = 42$$

$$3 < 4$$

$$k \leftarrow 4$$

$$a_4 \leftarrow 23 + 42x1 = \underline{65}$$

4 = 4 , stage 1 is complete

$$\ell \leftarrow 3$$

$$1 < 3$$

$$k \leftarrow 0$$

$$k \leftarrow 1$$

$$a_1 \leftarrow -1 + 1x1 = 0$$

$$1 < 3$$

$$k \leftarrow 2$$

$$a_2 \leftarrow -27 + 0x1 = -27$$

$$2 < 3$$


```
k ← 3
a3 ← 42 + -27x1 = 15
3 = 3 , stage one is complete
ℓ ← 2
1 < 2
k ← 0
k ← 1
a1 ← 0 + 1x1 = 1
1 < 2
k ← 2
a2 ← -27 + 1x1 = -26
2 = 2 , stage one is complete
ℓ ← 1
1 = 1
k ← 0
k ← 1
a1 ← 1 + 1x1 = 2
2 > 1 , stage one is complete
ℓ ← 0
1 > 0 , stage two is complete
END.
```

Our resultant polynomial is given as

$$P_4(X) = X^5 + 2X^4 - 26X^3 + 15X^2 + 65X - 78; X=x-1$$

Horner's method has become less popular in recent years as a practical numerical method of finding the real roots of a real algebraic polynomial equation. Gill

(1958), however, has constructed a very simple computer program which gives the real roots in binary form. The only arithmetic operations involved are addition, subtraction, and dividing by two. Convergence to a real root is assured although at a linear rate.

§ 2.31.10 Euclid's Algorithm for Two Polynomials in One Variable

Let $f(x)$ and $g(x)$ be the two polynomials

$$f(x) = a_0 x^n + \dots + a_{n-1} x + a_n,$$
$$g(x) = b_0 x^m + \dots + b_{m-1} x + b_m, \text{ where } n \geq m > 0.$$

By the greatest common divisor of these two polynomials is meant their common factor of greatest degree. Let us apply Euclid's algorithm to $f(x)$ and $g(x)$ precisely as we did to determine the remainder when a polynomial is divided by a linear factor. We thus get the system of identities

$$\begin{aligned} f(x) &\equiv Q_0(x) g(x) + R_1(x), \\ g(x) &\equiv Q_1(x) R_1(x) + R_2(x), \\ &\dots\dots\dots \\ R_{p-1}(x) &\equiv Q_p(x) R_p(x) + R_{p+1}. \end{aligned}$$

Then $q(x)$, $R_1(x)$, ... are polynomials of decreasing degrees, so that after a finite number of steps a remainder is reached which is a constant here indicated as R_{p+1} . Clearly, the necessary and sufficient condition that $f(x)$ and $g(x)$ be relatively prime is that $R_{p+1} \neq 0$. If $R_{p+1} = 0$ then $R_p(x)$ is the greatest common divisor of $f(x)$ and $g(x)$, Bocher (1922).

§ 2.31.11 Dialytic Elimination for Two Polynomials in One Variable

Let $f(x)$ and $g(x)$ be the two polynomials

$$f(x) = a_0 x^n + \dots + a_n,$$

$$g(x) = b_0 x^m + \dots + b_m, \text{ where } \underline{n} > \underline{m} > 0.$$

We multiply the first equation by the successive powers of x ,

$$x^{m-1}, x^{m-2}, \dots, x^2, x$$

and the second equation by

$$x^{n-1}, x^{n-2}, \dots, x^2, x$$

thus obtaining $n+m$ linear equations, homogeneous in $x^{n+m-1}, x^{n+m-2}, \dots, x, 1$. From the theory of linear equations we have that the necessary and sufficient conditions for the two polynomials in one variable to be relatively prime is that their resultant

$$R\left(\begin{smallmatrix} a_0, \dots, a_n \\ b_0, \dots, b_m \end{smallmatrix}\right) = \begin{vmatrix} a_0 & \dots & a_n & 0 & \dots & 0 \\ 0 & a_0 & \dots & a_n & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & a_0 & \dots & \dots & a_n \\ 0 & \dots & \dots & 0 & b_0 & \dots & b_m \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ b_0 & \dots & \dots & \dots & b_m & 0 & \dots & 0 \end{vmatrix}$$

does not vanish.

We define the i -th subresultant R_i of the two polynomials in one variable as the determinant obtained by

striking out the first and last 1 rows and columns from the resultant of these two polynomials. The degree of the greatest common divisor of $f(x)$ and $g(x)$ is then equal to the subscript of the first of the subresultants $R_0=R, R_1, R_2, \dots$ which does not vanish.

A more efficient representation for the resultant

$$R(a_0, \dots, a_n; b_0, \dots, b_m), \text{ where } n-m=r,$$

is possible which requires only an $n \times n$ determinant whose elements are the minors of matrix:

$$\begin{bmatrix} a_0 & a_1 & \dots & a_r & a_{r+1} & \dots & a_n \\ . & . & \dots & b_0 & b_1 & \dots & b_m \end{bmatrix}$$

The case for $n=3, m=2$ is given as

$$R(a_0, a_1, a_2, a_3; b_0, b_1, b_2) = \begin{vmatrix} |a_0 b_0| & |a_0 b_1| & |a_0 b_2| \\ |a_0 b_1| & (|a_0 b_2| + |a_1 b_1|) & |a_1 b_2| \\ |a_0 b_2| & |a_1 b_2| & |a_2 b_2| \end{vmatrix}.$$

A more extensive treatment of this subject may be found in Bocher (1922) and Turnbull (1944).

§ 2.32 Upper Bounds to the Real Roots of a Polynomial

In order to determine the real roots of a numerical equation it is advantageous to narrow the region within which they must be sought. In this section we shall concern ourselves with the problem of determining the upper bound to the positive roots of $P_n(x)$; the lower bound on the negative roots can be found in a similar manner.

§ 2.32.01 Lemma

An upper bound to the positive roots of

$$P_n(x) = x^n + a_1 x^{n-1} + \dots + a_n = 0$$

is given by

$$\sqrt[r]{|a_k| + 1},$$

where $a_r x^{n-r}$ is the first negative term and a_k is the largest negative coefficient.

Proof: Let x be such that

$$x^n > |a_k| (x^{n-r} + x^{n-r-1} + \dots + x + 1) > |a_k| \frac{x^{n-r+1} - 1}{x - 1}$$

and hence $P_n(x) > 0$.

Taking $x > 1$, the above inequality is satisfied by

$$x^n > |a_k| x^{n-r+1} / (x-1), \text{ or}$$

$$x^{n+1} - x^n > |a_k| x^{n-r+1}, \text{ or}$$

$$x^{r-1} (x-1) > |a_k|, \text{ which inequality is again}$$

satisfied by

$$(x-1)^{r-1} (x-1) \geq |a_k|,$$

since $x^{r-1} > (x-1)^{r-1}$. We have, therefore,

$$(x-1)^r \geq |a_k|, \text{ or } x \geq 1 + \sqrt[r]{|a_k|}.$$

§ 2.32.02 Lemma

If in any polynomial each negative coefficient is taken positively, and divided by the sum of all the positive coefficients which precede it, and if q is the greatest of these quotients, then $q + 1$ is an upper bound of the positive roots.

§ 2.33 Approximate Determination of Roots

The numerical solution of an arbitrary high degree polynomial is best affected in several steps. The first step consists of the approximate determination of the moduli of some or all of the roots by e.g. the root-squaring process. Higher accuracy if required can be achieved by an iterative procedure such as Newton's rule, for real zeros, and Bairstow's formula, for complex zeros. In this section we review the graphical, Bernoulli-Aitken, and Graeffe root-squaring methods for the approximate determination of roots. A more extensive account of these methods is available in Olver (1951).

An account of the numerical solution of algebraic equations using an electronic computer for Bernoulli's method, the root-squaring method, and the Newton-Raphson method is available in Brooker (1952). A more recent approach to the algebraic problem known as "direct search" is given in Ward (1957) and Lance (1959).

§ 2.33.01 The Graphical Method

The problem of determining the roots of $P_n(x) = 0$ can be reduced to the problem of graphing $P_n(x)$ in certain narrow regions. Information can be gained concerning complex roots by observing that a pair of complex roots exist at $x = x_0$ if

$$(i) \quad P_n'(x_0) = 0, \quad P_n''(x_0) > 0, \quad P_n(x_0) > 0,$$

or
$$(ii) \quad P_n'(x_0) = 0, \quad P_n''(x_0) < 0, \quad P_n(x_0) < 0.$$

The Graphical Method of solving for the roots of a polynomial can be readily extended to non-algebraic functions and also to the case of two polynomial equations in two unknowns.

§ 2.33.02 Graeffe's Root-squaring Process

The root-squaring process is essentially a means of calculating the moduli of the roots. The basis of this

method is that if the polynomial

$$P_n(x) = a_0 x^n + a_1 x^{n-1} + \dots + a_n,$$

has n simple zeros arranged as $|x_1| \geq |x_2| \geq \dots \geq |x_n|$,

then a polynomial whose zeros are $-x_1^2, -x_2^2, \dots, -x_n^2$,

is given by

$$P_n(x) = b_0 x^n + b_1 x^{n-1} + \dots + b_n,$$

where $b_s = a_s^2 - 2a_{s-1} a_{s+1} + 2a_{s-2} a_{s+2} - \dots$ ($s=0,1,\dots,n$).

The application of this transformation m times in succes-

sion yields a polynomial whose zeros are $-x_1^M, -x_2^M, \dots,$

$-x_n^M$, where $M = 2^m$. For sufficiently large m , the trans-

formed polynomial breaks up and the moduli of the zeros can

be computed from the ratios of the arithmetical M -th roots

of adjacent coefficients (see, e.g., Whittaker & Robinson,

1944, § § 54 to 58).

A major difficulty arising in hand computing applications of the root-squaring process is the checking for errors in each transformation; undetected mistakes can make subsequent computations useless. The only suitable check available is the unsatisfactory method of ordinary duplication. The computational arrangement of the calculations is very important and the technique of handling the coefficients in the form $N = p \times 10^q$ — where p lies between 1 and 10 and q is an integer — is recommended. Important difficulties which often occur with the root-squaring of high-degree polynomials are

(i) the failure of the polynomial to break up

after a resonable number of transformations,
(11) the severe loss of significant figures
because of cancellation.

§ 2.33.03 Aitken-Bernoulli Method

From the standpoint of speed and certainty of success the Aitken-Bernoulli process is inferior to Graeffe's root-squaring method. This method is useful, however, for evaluating one or two extreme zeros and it may be used on automatic computing equipment. A detailed account of this method is given by Aitken (1926).

The basis of Bernoulli's method is that if the polynomial

$$P_n(x):f(x) = a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n$$

has n simple zeros x_1, x_2, \dots, x_n (assumed arranged in the form $|x_1| \geq |x_2| \geq \dots \geq |x_n|$) then the difference equation

$$a_0f_1(t+n) + \dots + a_{n-1}f_1(t+1) + a_nf_1(t) = 0 \quad (1)$$

has the general solution

$$f_1(t) = w_1x_1^t + w_2x_2^t + \dots + w_nx_n^t,$$

where w_1, w_2, \dots, w_n are arbitrary constants. If $|x_1| > |x_2|$ then clearly

$$f_1(t) \sim w_1x_1^t \text{ and } f_1(t+1)/f_1(t) \rightarrow x_1, \text{ as } t \rightarrow \infty. \quad (2)$$

From a sequence of numerical values of $f_1(t)$ constructed by using (1) as a recurrence relation, the value of x_1 can

usually be obtained to any given degree of accuracy from (2), provided the sequence is taken far enough. The factor $x-x_1$ can be removed from $P_n(x)$ and the process repeated to obtain x_2 , provided $|x_2| > |x_3|$. Further applications enable $P_n(x)$ to be solved completely, provided its zeros are all real and distinct.

Aitken's generalization is designed to determine all the zeros from the sequence $\{f_1(t)\}$ and to include the cases of complex and multiple zeros. He constructs further sequences $\{f_s(t)\}$ defined by

$$f_2(t) = \begin{vmatrix} f_1(t) & f_1(t+1) \\ f_1(t-1) & f_1(t) \end{vmatrix}, \quad f_{s+1}(t) = \begin{vmatrix} f_s(t) & f_s(t+1) \\ f_s(t-1) & f_s(t) \end{vmatrix} \div f_{s-1}(t)$$

($s \geq 2$), and proves that if

$Z_s(t) \equiv f_s(t+1) / f_s(t)$ and $|x_s| > |x_{s+1}|$, then

$$Z_s(t) \rightarrow x_1 x_2 \dots x_s, \text{ as } t \rightarrow \infty. \quad (3)$$

This result allows us to evaluate numerically the real zeros and moduli of the complex zeros. We consider the case of x_1, x_2 real and distinct, x_3, x_4 a pair of complex conjugates $re^{\pm i\theta}$ and that $|x_2| > r > |x_5|$. Aitken proves that

$$Z_3(t) \sim x_1 x_2 r \cos \left\{ (t+1)\theta + \alpha \right\} \sec(t\theta + \alpha),$$

where α is an arbitrary constant, and hence that

$$\left\{ Z_3(t+1) Z_3(t) + k^2 \right\} / Z_3(t) \rightarrow 2k \cos \theta,$$

where $k \equiv x_1 x_2 r$ and is known by a previous application of (3).

A serious disadvantage of Aitken's process is the severe cancellation that occurs in the numerical calculation of the sequence $\{f_s(t)\}$. In fact, it can be shown

that if R leading figures of $Z_s(t)$ and $Z_s(t-1)$ are identical, then R figures are lost 'off the front' in forming $f_{s+1}(t)$. Since R increases with t , this cancellation becomes more and more severe as the work progresses and hence a large number of significant figures must be retained in the early sequences of the process.

§ 2.33.04 Whittaker's Theorem

The root of the equation

$$P_n(x) = a_0 x^n + \dots + a_n = 0$$

which is smallest in absolute value, is given by the series

$$\frac{-a_n}{a_{n-1}} - \frac{a_n^2}{a_{n-1} \begin{vmatrix} a_{n-1} & a_{n-2} \\ a_n & a_{n-1} \end{vmatrix}} - \frac{a_n^3}{a_{n-1} \begin{vmatrix} a_{n-1} & a_{n-2} \\ a_n & a_{n-1} \end{vmatrix}} \begin{vmatrix} a_{n-2} & a_{n-3} \\ a_{n-1} & a_{n-2} \end{vmatrix} - \dots,$$

provided that the series converges. There is no indication in the literature that this method has been successfully used in large-scale examples. Experience would suggest that methods based on the evaluation of determinants of large order is laborious and inefficient. A proof of Whittaker's Theorem is given in Whittaker and Robinson (1944) and in Turnbull (1944).

§ 2.34 Improving the Accuracy of Roots

Given some approximation to a zero, an iterative process if successful must produce a better approximation. Except in special cases (such as the well-known procedure for evaluating a square root), convergence is not guaranteed. Both the possibility and the speed of convergence depend in general upon the accuracy of the first approximation.

Iterative methods are therefore most useful when a fair approximation is already known. Iteration should be regarded as a supplementary process to some direct method such as root-squaring.

A measure of the power of an iterative process can be taken as the degree or order of convergence which is defined as follows:

If α is a simple or multiple zero of $f(x)$, a is an approximation to α and $\eta = \alpha - a$, then an iteration formula of the form

$$\eta = F(a) + O(|\eta|^k) \quad (k \geq 1)$$

is said to be convergent to the k th degree. We note at this point that we need not restrict this process to algebraic problems, Olver (1952).

§ 2.34.01 Method of False Position

If $f(x)$ has opposite signs at $x = x_1$ and $x = x_2$, then at least one root lies between x_1 and x_2 . Regarding x_1 and x_2 as approximations to the root, we can obtain a new approximation x_3 by inverse linear interpolation, according to the formula

$$x_3 = \frac{x_1 f(x_2) - x_2 f(x_1)}{f(x_2) - f(x_1)} = x_2 - \frac{(x_2 - x_1) f(x_2)}{f(x_2) - f(x_1)}.$$

This process can be repeated and is convergent if the function is continuous in $[x_1, x_2]$.

§ 2.34.02 Newton-Raphson Method

If x_1 is an approximation to a zero x_0 of $f(x)$ then in general a better approximation is $x_1 + \delta x_1$, where

$$\delta x_1 = - f(x_1) / f'(x_1).$$

This formula can be shown to be quadratically convergent. Two conditions which need to be satisfied in order that this method converge to a real root are

$$(i) f'(x) \neq 0 \text{ for } x \in [x_1, x_0], \quad \text{and}$$

$$(ii) f''(x) \neq 0 \text{ for } x \in [x_1, x_0],$$

Whittaker and Robinson (1944).

§ 2.34.03 Cubically Convergent Formulae

A simple formula is

$$\delta x_1 = - \frac{f(x_1)}{f'(x_1)} - \frac{(f(x_1))^2 f''(x_1)}{2(f'(x_1))^3}.$$

Another formula is that of Laguerre

$$\delta x_1 = - \frac{nf(x_1)}{f'(x_1) \pm ((n-1)^2 (f'(x_1))^2 - n(n-1)f(x_1)f''(x_1))}$$

where n is the degree of $f(x)$. This method amounts to approximating the polynomial by parabolas between two zeros, Todd (1962).

§ 2.34.04 Bairstow's Method

Let the results of dividing $P_n(x)$ twice in succession by $x^2 - px - \ell$ be denoted by

$$P_n(x) = (x^2 - px - \ell) P_{n-2}(x) + q_1x + (q_0 - pq_1),$$

$$P_{n-2}(x) = (x^2 - px - \ell) P_{n-4}(x) + T_1x + (T_0 - pT_1). \quad (1)$$

Then Bairstow's formula is given by

$$D\delta p = T_1q_0 - T_0q_1, \quad D\delta \ell = Mq_1 - T_0q_0,$$

where $M = lT_1 + pT_0$, $D = T_0^2 - MT_1$. (2)

Simple checks on the computation are given by

$$T_1 \delta l + T_0 \delta p = -q_1 , T_0 \delta l + M \delta p = -q_0 .$$

It can be shown that this method is quadratically convergent, M.C.M. (1961).

§ 2.35 Evaluation of Zeros of Ill-conditioned Polynomials

Let α be a simple zero of the general polynomial $P_n(x)$ given as

$$P_n(x) = x^n + a_1 x^{n-1} + \dots + a_n, \text{ i.e. } a_0 = 1.$$

Then if a_{n-m} is increased by δa_{n-m} , the value of $P_n(\alpha)$ changes from zero to $\alpha^{n-m} \delta a_{n-m}$, to the first order of approximation. Hence, by Newton's rule, the corresponding change in the zero is

$$\delta \alpha = - \alpha^{n-m} \delta a_{n-m} / P'_n(\alpha).$$

For the case of a double root we obtain

$$(\delta \alpha)^2 = - 2 \delta a_{n-m} \alpha^{n-m} / P''_n(\alpha).$$

The direct use of these formulae enable us to determine quite simply the limitations of accuracy that rounding or observational errors in the coefficients impose upon each real simple or double zero, Wilkinson (1959) and Olver (1951).

An example given in Wilkinson (1959) is reviewed here. The polynomial is of degree 20 and is defined by

$$P_{20}(x) = (x+1)(x+2)\dots(x+20).$$

Clearly,

$$P'_{20}(x) = \prod_{j \neq 1} (\alpha_1 - \alpha_j),$$

and

$$\delta\alpha_{20} = \frac{-20^{n-m} \delta a_{n-m}}{19!}.$$

This takes its greatest value for $n-m = 19$, namely

$$\delta\alpha_{20} = \frac{20^{19}}{19!} \delta a_1 \approx 0.43 \times 10^8 \delta a_1.$$

The 20th root is not the most sensitive to variations in a_{19} however. We have in fact

$$\delta\alpha_5 = \frac{15^{19}}{5!14!} \delta a_1 \approx 0.21 \times 10^{10} \delta a_1, \text{ and}$$

$$\delta\alpha_4 = \frac{16^{19}}{4!15!} \delta a_1 \approx 0.24 \times 10^{10} \delta a_1.$$

We observe that the multiplication factors are so large that for a change of order 10^{-7} in δa_1 , the linear approximation

$$\delta\alpha = -\alpha^{n-m} \delta a_{n-m} / P'_n(\alpha)$$

is completely invalid; e.g. a pair of complex roots of the new polynomial

$$P_{20}(x) = (x+1)(x+2)\dots(x+20) + 2^{-23}x^{19} \text{ are} \\ -16.730737466 \pm 2.812624894i$$

§ 2.4 Polynomials in Several Variables

Our review of the single-variable algebraic problem is now extended to the nonlinear algebraic problem in several variables. We concern ourselves with the case of n real variables x_1, x_2, \dots, x_n , denoted by $x:nx1$. A more extensive treatment of this problem is given in Bocher (1922).

§ 2.41 Theorems, Definitions, and Algorithms

§ 2.41.01 Definition

By a factor or divisor of a polynomial $f(x)$ is understood a polynomial $g(x)$ which satisfies an identity of the form

$$f(x) \equiv g(x) h(x) ; x : nx1 ,$$

$h(x)$ being also a polynomial.

§ 2.41.02 Definition

A real polynomial is said to be reducible in the domain of reals if it is identically equal to the product of two real polynomials neither of which is a constant.

§ 2.41.03 Definition

Two polynomials are said to be relatively prime if they have no common factor other than a constant.

Let $f(x_1, x_2)$ be any polynomial in two variables x_1, x_2 arranged according to powers of x_1 ,

$$f(x_1, x_2) \equiv a_0(x_2)x_1^n + \dots + a_n(x_2) = 0 ,$$

the a 's being polynomials in x_2 .

§ 2. 41.04 Theorem

A necessary and sufficient condition that a polynomial in x_2 alone, $g(x_2)$, be a factor of $f(x_1, x_2)$ is that it be a factor of all the a 's.

§ 2.41.05 Theorem

If two polynomials $f(x_1, x_2)$ and $g(x_1, x_2)$ are relatively prime, there is only a finite number of pairs of values of (x_1, x_2) for which f and g both vanish.

§ 2.41.06 Remainder Theorem for Polynomials in Two Variables

A necessary and sufficient condition that $f(x_1, x_2)$ and $g(x_1, x_2)$ have a common factor which involves x_1 is that $R(x_2) = 0$, where $R(x_2)$ is defined by Euclid's algorithm in § 2.41.08.

§ 2.41.07 Remainder Theorem for Polynomials in Several Variables

A necessary and sufficient condition that $f(x)$ and $g(x)$; $x: nx_1$, have a common factor which involves x_1 is that $R(x_2, \dots, x_n) = 0$.

Theorems § 2.41.04 and § 2.41.05 can also be readily extended to n variables simply by replacing y by the $n-1$ variables x_2, \dots, x_n . A more extensive treatment on this topic is given in Bocher (1922).

§ 2.41.08 Euclid's Algorithm of the Greatest Common Divisor for Polynomials in Several Variables

We consider the special case for polynomials in two variables. Let $f(x_1, x_2)$ and $g(x_1, x_2)$ be any two

polynomials in x_1 and x_2 arranged according to powers of x_1 ,

$$f(x_1, x_2) \equiv a_0(x_2) x_1^n + \dots + a_n(x_2), \quad a_0(x_2) \neq 0,$$

$$g(x_1, x_2) \equiv b_0(x_2) x_1^m + \dots + b_m(x_2), \quad b_0 \neq 0, n \geq m > 0.$$

Dividing $f(x_1, x_2)$ by $g(x_1, x_2)$ we obtain the identity

$$P_0(x_2) f(x_1, x_2) \equiv Q_0(x_1, x_2) g(x_1, x_2) + R_1(x_1, x_2).$$

We now define the computing algorithm.

```

1 ; R(x1, x2) ← P(x2)f(x1, x2) - Q(x1, x2)g(x1, x2)
    R(x1, x2): R(x2)                                     , → END
    f(x1, x2) ← g(x1, x2)
    g(x1, x2) ← R(x1, x2)
    → 1
    END

```

§ 2.42 Sylvester's Dialytic Method of Elimination

From n separate equations it is generally possible to eliminate $n-1$ unknowns. Sylvester's dialytic method provides a systematic means of eliminating one unknown from two equations. From a reverse point of view the vanishing of the resultant may be considered as the condition under which the two equations in question possess a common root.

We consider the case of two equations in two unknowns; i.e. let $f(x_1, x_2)$ and $g(x_1, x_2)$ be any two polynomials in x_1 and x_2 arranged according to powers in x_1 ,

$$f(x_1, x_2) \equiv a_0(x_2) x_1^n + \dots + a_n(x_2),$$

$$g(x_1, x_2) \equiv b_0(x_2) x_1^m + \dots + b_m(x_2), \text{ where } n > m > 0.$$

We multiply the first equation by the successive powers of x_1 ,

$$x_1^{m-1}, x_1^{m-2}, \dots, x_1,$$

and the second by

$$x_1^{n-1}, x_1^{n-2}, \dots, x_1,$$

thus obtaining $n + m$ linear equations, homogeneous in

$x_1^{n+m-1}, \dots, x_1, 1$. The condition for their consistency is, by the theory of linear equations,

$$R \equiv \begin{vmatrix} a_0(x_2) & \dots & a_n(x_2) & 0 & \dots & 0 \\ 0 & a_0(x_2) & \dots & a_n(x_2) & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & b_0(x_2) & \dots & b_m(x_2) & 0 & 0 \\ b_0(x_2) & \dots & b_m(x_2) & 0 & \dots & 0 \end{vmatrix} = 0$$

The determinant $R = 0$ can also be written as $P(x_2) = 0$.

The roots of this polynomial are possible solutions for x_2 of the simultaneous system. The values of x_1 can then be found by substitution.

From three equations, consistent in two unknowns x_1 and x_2 , we could eliminate x_1 from the first and second, and again from the first and third equation, giving two resultants containing x_2 , from which x_2 could be eliminated by a like procedure. We note, however, that the total degree of the resultant polynomial can become very large.

§ 2.42.01 Example of Sylvester's Dialytic Method

Find the simultaneous solution for

$$x^2 + y^2 - y = 1$$

$$x y = 1.$$

using the Dialectic method we obtain

$$\begin{pmatrix} 1 & 0 & y^2-y-1 & x^2 \\ 0 & y & -1 & \\ y & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 0$$

For consistency we must have

$$P(y) = \begin{vmatrix} 1 & 0 & y^2-y-1 \\ 0 & y & -1 \\ y & -1 & 0 \end{vmatrix} = 0 ,$$

i.e. $-1 + (y^2-y-1)(-y^2) = 0$, or

$$P_4(y) = y^4 - y^3 - y^2 + 1 = 0 .$$

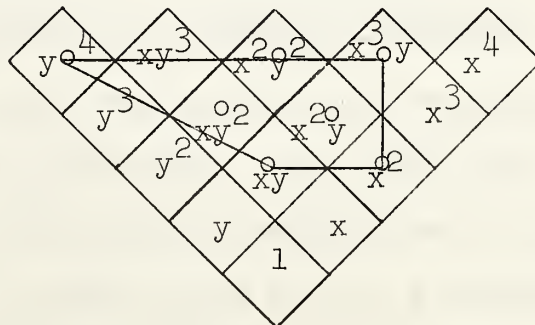
This fourth degree polynomial has at most two real roots by Descartes Rule of Signs.

§ 2.5 Curve Tracing of Polynomials in Two Variables

The graph of a function may in some cases be an important aid in the solution of an algebraic problem. Although the graph cannot be said to represent a function accurately, nevertheless, it is a useful technique for exhibiting both the nature and approximate value of a solution. The special techniques available for graphing a function in two variables are not well-known and hence the following review is included in this thesis. Specifically, we will discuss the properties and use of the analytic triangle.

§ 2.51 De Gua Triangle

We construct the following diagram and label it as shown.

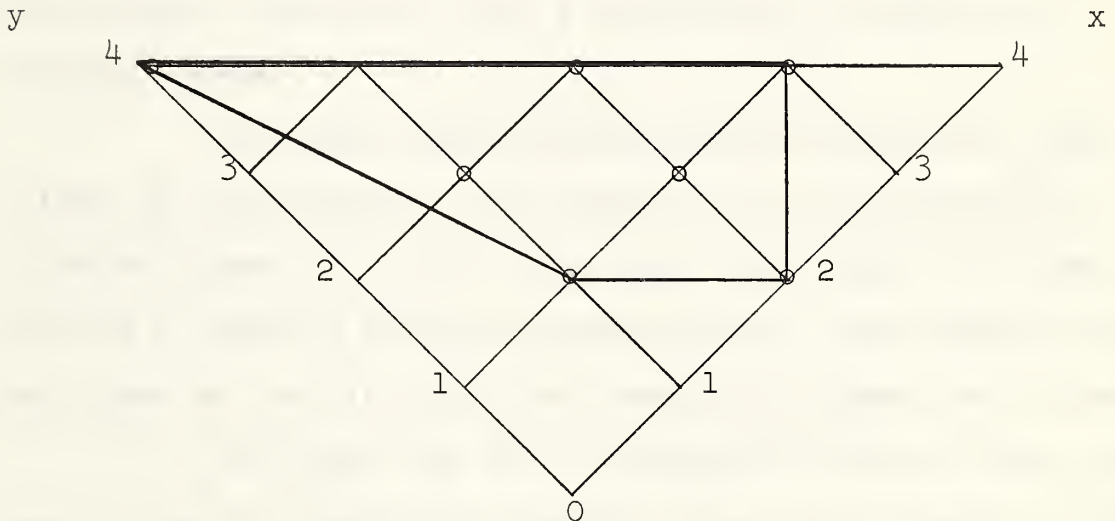


An equation is said to be placed on the triangle if a cross, or dot is placed on the center of the rectangle corresponding to a term of the equation. In the diagram shown we have placed the equation

$$f_4(x,y) = ay^4 + bx^2y + cx^3y + dxy^2 + ex^2 + fxy + gx^2y^2.$$

We then proceed to form a convex polygon exterior to which no point lies.

Another triangle scheme employed is as follows.



We place the same equation on this analytic triangle as for the De Gua triangle. The analytic triangle is more useful since fractional indices - hence the name analytic - can be placed on the triangle more easily.

§ 2.52 Properties of the Analytic Triangle

When all the terms of an equation are placed on the triangle, the following properties hold, with respect to any straight line which contains two or more circles.

(1) If every term not on a straight line is rejected the resulting equation gives one or more constant

values of the ratio $y^s : x^r$, where the values of r and s depend only on the direction of the line.

(2) If the extended straight line intersects both sides of the triangle or the extended triangle with the relation $y^s : x^r$ the terms of the original equation on the same side of the line as the right angle will be less and the terms on the other side of the line will be greater than any of the terms on the line for x and y very large; the reverse holds for x and y very small if there is no constant term.

(3) When the line makes acute angles with both sides of the triangle, the relation is of the form $y^s x^r =$ constant; when y is very large and x very small the terms on the y side are greater, those on the x side smaller than any term on the line and vice versa for x great and y small.

(4) When the line is parallel to one of the sides, say to Ox , the resulting equation gives one or more straight lines which are parallel to Oy ; and when y is very large each term on the side of O is less and those terms on the other side are greater than the terms on the line.

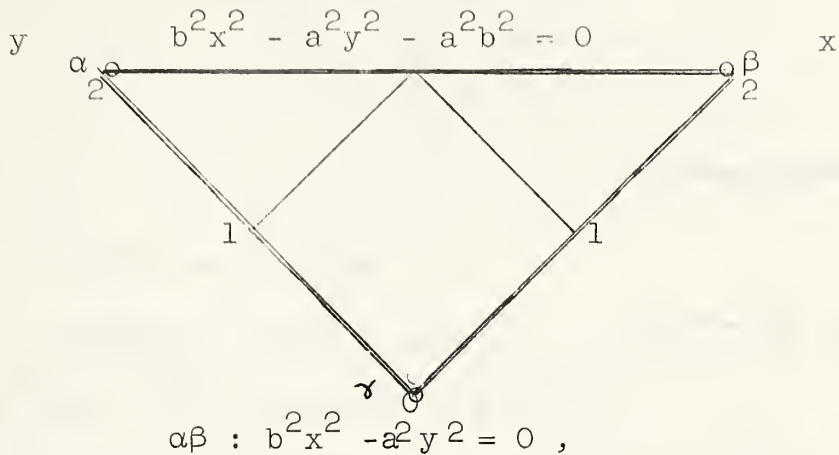
(5) When the line coincides with a side of the triangle the resulting equation defines the points of intersection with the corresponding axis.

Furthermore if any line gives a first approximation to an asymptote, the terms to be taken into account for the second approximation are found by moving the line parallel to itself until it passes through another dot or group of dots, all of which correspond to the terms required to be taken

into account.

§ 2.53 Examples of the Method

§ 2.53.01 Example I



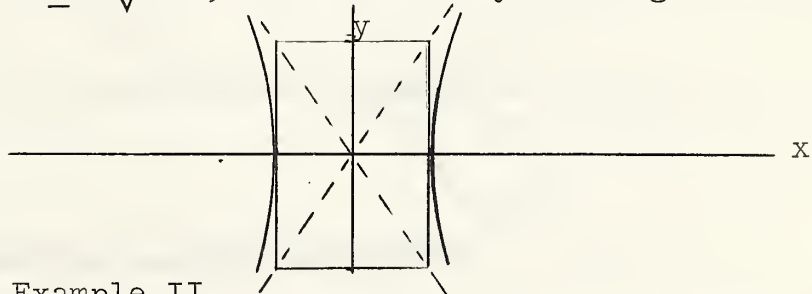
or $y = \pm \frac{b}{a} x$, the usual expression for the asymptotic lines of an hyperbola.

$$\beta\gamma : b^2x^2 - a^2b^2 = 0 ,$$

or $x = \pm a$, the intercepts for the hyperbola.

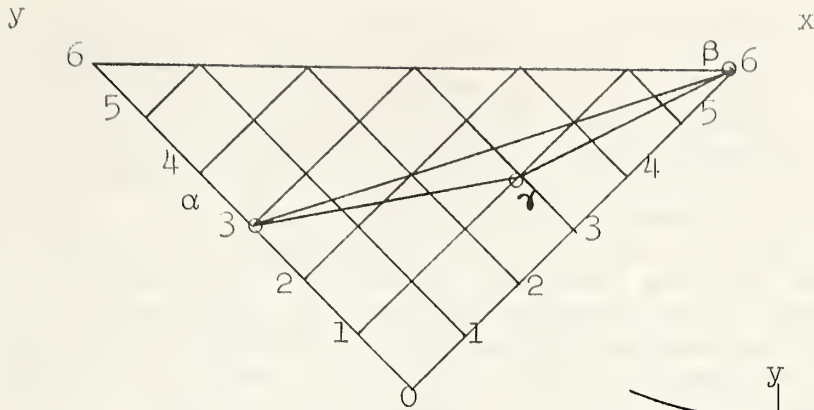
$$\gamma\alpha : -a^2y^2 - a^2b^2 = 0$$

$y = \pm b\sqrt{-1}$, the auxiallary rectangle dimension.



§ 2.53.02 Example II

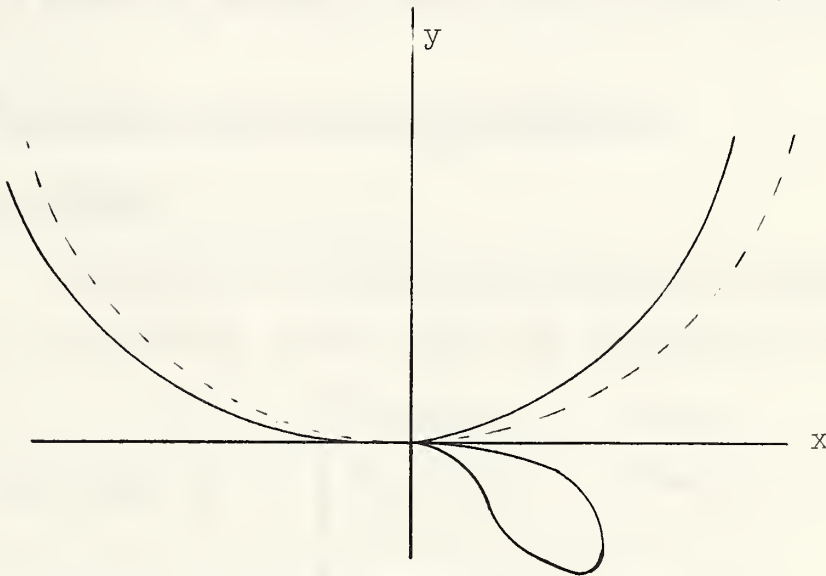
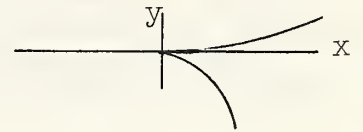
$$x^6 + 2a^2x^3y - b^3y^3 = 0$$



$$\alpha\beta: x^6 - b^3y^3, \text{ or } x^2 - by = 0$$

$$\beta\gamma: x^6 + 2a^2x^3y, \text{ or } x^3 + 2a^2y = 0$$

$$\gamma\alpha: 2a^2x^3y - b^3y^3, \text{ or } 2a^2x^3 - b^3y^2 = 0$$



§ 2.6 Systems of Linear Equations

An important technique in solving systems of nonlinear equations is the method of iteration. The merit of the iterative or approximate schemes is that they employ simple and uniform operations and hence are easily mechanized. Before we proceed to the application of iterative methods to nonlinear systems we find it convenient to consider first the theoretical and computing aspects of linear systems. In this way we encounter in a simple context most of the matrix properties we require.

The problem of solving a non-homogeneous linear algebraic system is closely related to the problem of inverting a matrix and the problem known as elimination. Numerical methods of solving the stated problem are divisible into two groups: exact and iterative methods. The fundamental method which is referred to as an exact method is the Gauss method of elimination. The Gauss method is reviewed in this thesis although it has been superseded by the more compact and efficient schemes such as the square-root and L-U methods. Iterative methods which in general require only simple, uniform operations and hence are easily mechanized afford a means by which a system of linear equations may be solved approximately. Before we proceed with methods of solution we review the more basic definitions, notation, and theorems of linear algebra. The elementary accounts can be found in Fadeeva (1959) and Hildebrand (1952).

Further references on the topic of linear algebra include a treatment at an intermediate level by Fox, Husky, and Wilkinson (1948); a study on relaxation methods by Temple (1938); and a comprehensive work with an extensive bibliography by Forsythe (1953). A more recent account of iterative, accelerated iterative, gradient, and conjugate gradient methods is given in Hestenes and Stiefel (1952) and Martin and Tee (1961).

§ 2.61 Theorems, Definitions, and Notation

§ 2.61.01 Matrix

A matrix is defined as an array of numbers, complex in the general case, which can be written in the form:

$$A \equiv [a_{ij}] \equiv \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix},$$

consists of m rows and n columns of elements, where in the typical element a_{ij} , the first subscript (here i) denotes the row and the second subscript (here j) the column occupied by the element.

§ 2.61.02 Trace of a Matrix

The quantity $a_{11} + a_{22} + \dots + a_{nn}$ is called the trace of the matrix A and is denoted by Trace A.

§ 2.61.03 Transpose of a Matrix

The transpose of a matrix A denoted by A^T is obtained by interchanging the rows and columns of A, i.e.

$$A^T = [a_{ij}]^T = [a_{ji}]$$

§ 2.61.04 Inverse of a Matrix

A matrix B is called the inverse of a matrix A denoted by A^{-1} if

$AB = BA = I$, or $AA^{-1} = A^{-1}A = I$ where I is the identity matrix

$$I = [\delta_{ij}] .$$

§ 2.61.05 Adjoint of a Matrix

A matrix B is the adjoint matrix of A if

$$B = [A_{ji}]$$

where A_{ji} is the signed determinant obtained by striking out the i th row and j th column of A and multiplying by $(-1)^{i+j}$.

The adjoint matrix B satisfies the equation

$$AB = |A| I$$

and is related to the inverse matrix A^{-1} by

$$A^{-1} = B / |A| , \text{ where } |A| \neq 0.$$

§ 2.61.06 Matrix Product

The multiplication of two matrices A and B is

denoted by

$$C = AB$$

where C is a matrix whose elements c_{ij} are given by

$$c_{ij} = a_{i\ell} b_{\ell j}, \text{ summed over } \ell.$$

§ 2.61.07 Matrix Norms

1. $||A||_I = \max_i \sum_{k=1}^n |a_{ik}|$
2. $||A||_{II} = \max_k \sum_{i=1}^n |a_{ik}|$
3. $||A||_{III} = \left(\sum_{i,k=1}^n a_{ik}^2 \right)^{\frac{1}{2}} = (\text{Trace } A^T A)^{\frac{1}{2}}$
4. $||A||_{IV} = \max ((Ax)^T Ax)^{\frac{1}{2}} \text{ for } x^T x = 1$

The above norms will be referred to as I, II, III, IV, respectively. Norms I, II, III are applicable to either square or rectangular matrices but norm IV is applicable only to a square matrix.

All four norms satisfy the following conditions:

- (1) $N(A) > 0$ if $A \neq 0$ and $N(A) = 0$ iff. $A \equiv 0$;
- (2) $N(cA) = |c| N(A)$ where c is a scalar;
- (3) $N(A+B) \leq N(A) + N(B)$;
- (4) $N(AB) \leq N(A) N(B)$;
- (5) matrix norms I, II, and III are upper bounds to the eigenvalues of any matrix.

§ 2.61.08 Theorem 1

A necessary and sufficient condition that $A^m \rightarrow 0$ is that the matrix A has eigenvalues λ_i such that $|\lambda_i| < 1$, all i.

Proof

Let A be brought into diagonal form i.e.
 $A = P \Lambda P^{-1}$ where $\Lambda_{ij} = \lambda_i \delta_{ij}$. Then $A^m = P \Lambda^m P^{-1}$
 where $\Lambda_{ij}^m = \lambda_i^m \delta_{ij}$.

Hence for $A^m \rightarrow 0$ it is necessary and sufficient that $\Lambda^m \rightarrow 0$,
 for which it is in turn necessary and sufficient that $|\lambda_i| < 1$,
 all i . In the case the matrix A cannot be brought into
 diagonal form, the theorem is proved either with the aid of
 considerations of continuity or by passing to the Jordan
 canonical form.

§ 2.61.09 Theorem 2

In order that $A^m \rightarrow 0$, it is sufficient that any
 one of the norms of A be less than unity.

Proof

Now $\|A^m\| \leq \|A^{m-1}\| \|A\| \leq \dots \leq \|A\|^m$.

Therefore $\|A^m\| \rightarrow 0$ if $\|A\| < 1$ and thus $A^m \rightarrow 0$

§ 2.61.10 Theorem 3

The necessary and sufficient conditions that the
 series

$$I + A + A^2 + \dots + A^m + \dots$$

be convergent is that $A^m \rightarrow 0$ as $m \rightarrow \infty$.

Proof

From a previous theorem we have $|I-A| \neq 0$ and
 therefore $(I-A)^{-1}$ exists.

Consider

$$(I + A + A^2 + \dots + A^k)(I-A) = I - A^{k+1}.$$

Postmultiplying by $(I-A)^{-1}$ we have

$$I + A + A^2 + \dots + A^k = (I-A)^{-1} - A^{k+1} (I-A)^{-1}$$

and therefore

$$\text{L.H.S.} \rightarrow (I-A)^{-1} \quad \text{since } A^{k+1} \rightarrow 0.$$

§ 2.61.11 Theorem 4

An estimate of the rate of convergence of the series of the previous theorem for

$$||A|| < 1 \text{ is } \frac{||A||^{k+1}}{1 - ||A||}.$$

Proof

We have

$$(I-A)^{-1} - (I+A+\dots+A^k) = A^{k+1} + A^{k+2} + \dots$$

$$\text{hence } ||\text{L.H.S.}|| \leq ||A||^{k+1} + \dots = \frac{||A||^{k+1}}{1 - ||A||}.$$

§ 2.62 Definition of the Problem of Linear Algebra

A general set of n linear simultaneous algebraic equations in n unknowns x_1, x_2, \dots, x_n can be formulated as:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + \dots + a_{2n}x_n = b_2$$

$$\dots$$

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n$$

which can be written more compactly in matrix notation as

$$Ax = b ; A : nxn ; x, b : n \times 1.$$

The necessary and sufficient condition that $Ax = b$ have a unique solution is that $|A| \neq 0$.

§ 2.63 Direct Methods

§ 2.63.01 Cramer's Rule

If A is nonsingular then we may write the solution of our system of linear equations as

$$x = A^{-1} b = \frac{B}{|A|} b$$

where B is the adjoint of A .

$$\text{Hence } x_i = \frac{A_{1i}b_1 + A_{2i}b_2 + \dots + A_{ni}b_n}{|A|}$$

where A_{ki} are the cofactors of the element a_{ki} in the determinant of the matrix A . If we define

$$A_{1i}b_1 + A_{2i}b_2 + \dots + A_{ni}b_n = |A_i|$$

we may denote our solution as

$$x_i = \frac{|A_i|}{|A|} \quad (\text{Cramer's Rule}).$$

This result serves mainly as an existence theorem; it has no practical value.

§ 2.63.02 Gauss Method

The problem of obtaining the simultaneous solutions of n linear equations in n unknowns i.e. $Ax = b$ may be reduced to a problem of solving $n-1$ linear equations in $n-1$ unknowns by the simple method of elimination. By adding a suitable multiple of the first equation to all other equations so that in each resulting equation the coefficient

§ 2.64 Iterative Methods

§ 2.64.01 Simple Iteration

The linear problem $Ax = b$ may be written as $(I - L - U)x = b$ where $-L$, $-U$ are the lower and upper parts, respectively, of A which has been normalized so that the diagonal components of A are unity.

Now $x = (L + U)x + b$

$= Mx + b$ where $M = L + U$.

Choosing a vector x^0 arbitrarily we construct the sequence of vectors

$$x^1 = Mx^0 + b$$

$$x^2 = Mx^1 + b = M^2x^0 + (I+M)b$$

.....

$$x^{(k+1)} = M^{(k+1)}x^0 + (I+M+M^2+\dots+M^{(k-1)})b.$$

If M has eigenvalues such that $|\lambda_i| < 1$ for all i we have by Theorems 1 and 3 that

$$x^{(k+1)} \rightarrow (I - M)^{-1}b = A^{-1}b.$$

Hence $x = \lim_{k \rightarrow \infty} x^k = A^{-1}b$ which is the solution of the system

$$Ax = b.$$

We may use any norm $N(A)$ as a test for convergence (theorem 2) and a bound on the rate of convergence is given by

$$\frac{(N(A))^{k+1}}{1 - N(A)} \quad (\text{theorem 4}).$$

§ 2.64.02 Gauss Seidel Iteration

The linear problem $Ax = b$ may be written in the form $(I - L - U)x = b$ where I, L, U are as before.

Rewriting the above as $(I - L)x = Ux + b$ and iterating as follows:

$$(I - L)x^{k+1} = Ux^k + b .$$

Premultiplying by $(I - L)^{-1}$ we obtain

$$x^{k+1} = (I - L)^{-1}Ux^k + (I - L)^{-1}b$$

assuming $(I - L)$ is nonsingular.

$$\text{Let } (I - L)^{-1}U = C \text{ and } (I - L)^{-1}b = a$$

and obtain

$$\begin{aligned} x^{k+1} &= a + Cx^k \\ &= C^k x^0 + (I + C + C^2 + \dots + C^{k-1}) a \end{aligned}$$

where x^0 is an arbitrary vector.

As before if $(I - L)^{-1}U = C$ has eigenvalues such that $|\lambda_i| < 1$, all i , we have by Theorems 1 and 3 that

$$x^{k+1} \rightarrow (I - C)^{-1}a = A^{-1}b.$$

§ 2.64.03 Example I

Find x_1, x_2 for

$$.1x_1 + 1x_2 = 1$$

$$1x_1 + 2x_2 = 1$$

$$\text{Now } A = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}, \quad I - L = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}, \quad U = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

$$(I - L)^{-1} = \begin{pmatrix} 1 & 0 \\ -.5 & .5 \end{pmatrix}, \quad C = (I - L)^{-1}U = \begin{pmatrix} 0 & -.1 \\ 0 & .5 \end{pmatrix}, \quad a = (I - L)^{-1}b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Now C has eigenvalues 0, .5 and therefore the Gauss Seidel process will converge for arbitrary b .

$$\text{Let } b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, x^0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ c = \begin{pmatrix} 0 & -1 \\ 0 & .5 \end{pmatrix}, a = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

$$\text{Then } x^1 = a + cx^0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 \\ .5 \end{pmatrix} = \begin{pmatrix} 0 \\ .5 \end{pmatrix}$$

$$x^2 = a + cx^1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} -.5 \\ .25 \end{pmatrix} = \begin{pmatrix} 1-2^{-1} \\ 2^{-2} \end{pmatrix}$$

.....

$$x^k = \begin{pmatrix} 1-2^{-k+1} \\ 2^{-k} \end{pmatrix} \text{ so that}$$

$$x^k \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ as } k \rightarrow \infty.$$

§ 2.65 Gradient and Conjugate Gradient Iterative Methods

Again, we consider the linear system $Ax = b$ where A is a real symmetric positive definite matrix whose inverse A^{-1} is therefore known to exist. b and x are column vectors; the former is given, and it is required to calculate the elements of the unique solution

$$x = A^{-1}b.$$

We turn now to a different approach and consider the iterative methods which require the quadratic functional

$$f(x) = (x - A^{-1}b)^T A(x - A^{-1}b) \\ = x^T Ax - 2x^T b + b^T A^{-1}b.$$

The equations $f(x) = \text{constant}$ represent similar and similarly situated ellipsoids with center at $A^{-1}b$. Clearly, $f(x)$ takes its minimum value, zero, at $x = A^{-1}b$.

Any estimate x^k of $A^{-1}b$ defines a point in n -space and also the ellipsoid through this point. Approach to the center is represented by the sequence of vector displacements

$$x^{k+1} - x^k = \Delta x^k = t_k p^k$$

where t_k is a scalar and p^k a vector. We define a residual vector r^k as the inward pointing normal at x^k to the ellipsoid through x^k since

$$r^k = b - Ax^k = -\frac{1}{2} \nabla f(x^k).$$

The direction defined by r^k is that direction for which $f(x^k)$ decreases most rapidly and hence is known as the direction of optimum gradient or steepest descent. An iteration which is defined by a recursion of the form

$$x^{k+1} = x^k + \sum_{i=0}^k c_{ki} r^i$$

is called a gradient method.

In the succeeding sections we consider two variants of the gradient method.

§ 2.65.01 The Gradient Method

The direction at $x = x^k$ for which $f(x^k)$ decreases most rapidly is given by the negative gradient r^k . We continue in this direction of steepest descent until we arrive at the point x^{k+1} which satisfies the condition

$$(r^{k+1})^T r^k = 0 .$$

Using the definition

$$\begin{aligned} x^{k+1} - x^k &= t_k p^k , \text{ and} \\ r^k &= b - Ax^k , \text{ we obtain} \\ r^{k+1} &= r^k - t_k A p^k . \end{aligned}$$

Further, the optimum value of t_k for the direction p^k is

$$t_k^o = (p^k)^T r^k / (p^k)^T A p^k .$$

Alternately, differentiation shows that

$$f(x^k + t_k p^k) = t_k^2 (p^k)^T A p^k - 2t_k (p^k)^T r^k + f(x^k)$$

takes its minimum value for t_k^o .

The particular iteration defined by

$$x^{k+1} = x^k + t_k r^k$$

is the optimum gradient or steepest descent method of Temple (1939). Unfortunately, its convergence is usually too slow for the method to be of practical value. This method is of interest, however, since it serves as the basis of a large class of iterative methods.

§ 2.65.02 Conjugate Gradient Method

A better method based on the knowledge that the center of an ellipsoid lies on the plane conjugate to a given chord, known as the conjugate gradient method, can be most easily described by the following computing algorithm.

```

        x0 arbitrary
        r0 ← b - Ax0
        p0 ← r0
        k ← 0
1;   tk ← (rk)T pk / (pk)T A pk
        xk+1 ← xk + tk pk
        rk+1 ← rk - tk A pk
        sk ← -(rk+1)T A pk / (pk)T A pk
        pk+1 ← rk+1 + sk pk
        k ← k+1
        k : n                                , ≤ 1
END
    
```


The relation

$$(p^{k+1})^T A p^k = 0$$

is satisfied for all k , i.e., the directions p^k are A -conjugate to each other. In theory the process terminates after n steps but the inevitable rounding errors in a practical computation perturb the strategy. A more extensive treatment of this subject may be found in Hestenes and Steifel (1952) and Martin and Tee (1961).

CHAPTER III

TECHNIQUES FOR SOLVING NONLINEAR SYSTEMS

§ 3.1 Introduction

The problem of solving systems of nonlinear equations has been relatively neglected in the mathematical literature. The nature of the solutions and the possible pitfalls in any general method now available need more careful attention and elucidation.

For any linear system of the form

$$r = b - Ax ; A: nxn ; r, b, x: n \times 1 ,$$

there is a corresponding nonlinear problem of obtaining the solutions, if any exist, or the level points of the test function $T(x) = r^t C r ; C : nxn ,$ positive definite. The single function $T(x)$ will have a solution and vanish there, if, and only if $x = \alpha$, where α is a solution of the linear system. The properties of the original linear system are closely linked with the properties of the Hessian matrix

$$H = [\partial^2 T / \partial x_i \partial x_j] ,$$

which has constant elements. If $|H| \neq 0$, a unique level point of $T(x)$ will be a solution if the initial system is consistent. If $|H| = 0$, there will be a continuous locus of level points that, depending on definition, can be treated as infinitely many solutions or as no solution.

In the nonlinear as in the linear case, the Hessian matrix H - in general, a function of the independent variables

x_1, x_2, \dots, x_n — of a test function $T(x)$; $x : n \times 1$, generally characterizes the nature of the solution; in the nonlinear case, however, there are additional problems. The test function $T(x)$ most commonly used in connection with the nonlinear system $f(x) = 0$; $f(x) : n \times 1$, is defined as $T(x) = (f(x))^T f(x)$. The nonlinear test function $T(x)$ may have a unique level point; but it is more likely that several level points or relative level points may exist. Level points may be distinct or finitely confluent, may occur in continuous sets or any combination of these situations may exist in a single problem. For level points which are distinct we have $|H| \neq 0$ in some open region containing each level point. If $|H| = 0$ we have a confluent set of points and if $|H| \approx 0$ there may be a set of nearly coincident level points.

Milne (1949) has described a process by which the difficulty of coincident or nearly coincident level points may be resolved. The various types of simple level points can be distinguished by examining the quadratic form $x^T H x$ which is positive definitive only at a relative or absolute minimum solution of $T(x) = 0$.

Once a solution has been found, say $x = \alpha_0$, a further solution can be obtained from the successor problem:

$$T_1(x) = T(x) / ||x - \alpha_0||^p$$

where $||x||$ is a suitable norm of x and p a discrimination factor. Since it is impossible to determine an infinite number of solutions, limits on either the number of solutions or the range of their locus are always required.

The techniques reviewed in this study of the nonlinear problem usually require that $T(x)$ and the gradient vector $\nabla T(x)$ be calculated at the set of points $x_1^k, x_2^k, \dots, x_n^k$; from the behaviour of the function at this point in n -space, we obtain information concerning the location of possible solutions. The test points are chosen in a sequential manner by a fixed set of operations. The most recent literature on this topic is reviewed in Turner (1960), Spang (1962), Todd (1962), and Beckenbach (1956).

§ 3.2 Definitions and Theorems

§ 3.21 Definitions and Notation

§ 3.21.01 The nonlinear system is defined as $f(x) = 0$; $f(x), x : nx1$. The determinant $\left| \frac{\partial f_i(x)}{\partial x_j} \right|$ is called the Jacobian or Functional Determinant of the functions f_1, f_2, \dots, f_n with respect to the variables x_1, x_2, \dots, x_n and is denoted by

$$J = \frac{\partial(f_1, f_2, \dots, f_n)}{\partial(x_1, x_2, \dots, x_n)} = \frac{\partial f_i(x)}{\partial x_j}.$$

The Jacobian matrix $J = \left[\frac{\partial f_i(x)}{\partial x_j} \right]$ will also be referred to as the Jacobian.

§ 3.21.02 The Jacobian of the first differential coefficients of a function of n variables, taken with respect to the several variables, is called the Hessian of the function.

Thus if,

$$T = T(x_1, x_2, \dots, x_n),$$

$$\text{then } H = \frac{\partial(\frac{\partial T}{\partial x_1}, \frac{\partial T}{\partial x_2}, \dots, \frac{\partial T}{\partial x_n})}{\partial(x_1, x_2, \dots, x_n)} = \left[\frac{\partial^2 T}{\partial x_i \partial x_j} \right] \text{ is}$$

the Hessian of the function T .

§ 3.21.03 α is an exact solution of the nonlinear system $f(x) = 0$; $f(x)$, $x : n \times 1$, if $f(\alpha) = 0$; x^k is an approximate solution if $f(x^k) \approx 0$.

§ 3.21.04 The gradient vector of $T(x)$, where $T(x)$ is a function in n variables x_1, \dots, x_n , is given by :

$$g(x) = \nabla T(x) = \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \dots, \frac{\partial}{\partial x_n} \right) T(x); g(x) : n \times 1.$$

§ 3.21.05 A homogeneous polynomial of the second degree in several variables x_1, \dots, x_n is called a quadratic form.

Any real quadratic form can be written as:

$\sum_{i,k=1}^n a_{ik} x_i x_k$, where $a_{ik} = a_{ki}$, or in matrix notation as $x^T A x$, where $A^T = A$.

§ 3.21.06 A function in n variables denoted by $C(x)$; $x : n \times 1$, is convex in a domain D if for $x^1, x^2 (x^1, x^2 : n \times 1) \in D$ and any t , $0 < t < 1$ we have

$$C(tx^1 + (1-t)x^2) \leq t C(x^1) + (1-t) C(x^2).$$

A function is strictly convex if, for those values of t , the \leq sign can always be replaced by a $<$ sign.

§ 3.22 Theorems and Lemmas

§ 3.2.01 Theorem

The necessary and sufficient conditions that $C(x) = 0$ have a solution at α is that $g(\alpha) \equiv \nabla C(\alpha) = 0$ and $C(x)$ is a convex function, Curry (1944).

§ 3.22.02 Lemma

A positive semidefinite quadratic form is convex and a positive definite form is strictly convex.

Proof: Let $f(x) = x^T A x$, where $A = A^T$.

$$\text{Now } f(tx^1 + (1-t)x^2) = t^2(x^1)^T A x^1 + 2t(1-t)(x^1)^T A x^2 + (1-t)^2(x^2)^T A x^2,$$

$$\text{and } tf(x^1) + (1-t)f(x^2) = t(x^1)^T A x^1 + (1-t)(x^2)^T A x^2$$

$$\text{Hence } tf(x^1) + (1-t)f(x^2) - f(tx^1 + (1-t)x^2) = t(1-t)(x^1 - x^2)^T A (x^1 - x^2).$$

§ 3.22.03 Lemma

A square and symmetric matrix A is positive definite if $(s^T) A s > 0$ holds for $s \neq 0$, positive semidefinite if $(s^T) A s \geq 0$ for $s > 0$.

§ 3.3 Simple and Gauss-Seidel Iteration

The iterative methods which solve the linear problem $Ax=b$ can often be extended to solve the nonlinear system $f(x)=0$; $A:n \times n$; $x, b, f(x) : n \times 1$.

§ 3.31 Simple Iteration

If the initial nonlinear system $F(x) = 0$;
 $F(x) : n \times 1$, can be written in the special vector form
 $x = f(x)$; $f(x) : n \times 1$ we can construct a sequence of vectors
 by the iterative equation:

$$x^{k+1} = f(x^k) , x^0 \text{ arbitrary.}$$

The conditions required to ensure convergence to a solution
 are that $x^k \approx \alpha$, where the degree of proximity required depends
 on the functions $f_1(x), \dots, f_n(x)$, and all the eigenvalues
 of the Jacobian $J(\alpha)$ have moduli < 1 . A reasonable approxima-
 tion to this eigenvalue problem is given by

$$|\lambda \max_i| \leq \sum_{k=1}^n \frac{\partial f_i}{\partial x_k}(x) , i = 1, 2, \dots, n.$$

The example of the next section illustrates this point.

§ 3.32 Example of Simple Iteration

Let the problem be of the general form

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f_1(x,y) \\ f_2(x,y) \end{pmatrix} .$$

We consider the particular case

$$x = 0.2y^3 - 0.6y^2 - 0.6x^2y - 0.8xy + 0.6x^2 + 1.2x + 0.2y + 0.4$$

$$y = 0.2x^3 + 0.4x^2 - 0.6xy^2 + 1.2xy - 0.4y^2 + 1.2y - 0.2x + 0.8.$$

$$J(x,y) = \begin{pmatrix} -1.2xy - 0.8y + 1.2x + 1.2 & 0.6y^2 - 1.2y - 0.6x^2 - 0.8x + 0.2 \\ 0.6x^2 + 0.8x - 0.6y^2 + 1.2y - 0.2 & -1.2xy + 1.2x - 0.8y + 1.2 \end{pmatrix}$$

$$J(0.2, 1.3) = \begin{pmatrix} 0.1 & -0.5 \\ 0.5 & 0.1 \end{pmatrix} ; |\lambda \max| \leq 0.6 .$$

$$x^1 = 0.2(1.3)^3 - 0.6(1.3)^2 - 0.6(0.2)^2(1.3) - 0.8(0.2)(1.3) + 0.6(0.2)^2 + 1.2(0.2) + 0.2(1.3) + 0.4 = 0.11 ,$$

$y^1 = 1.77$, and so on. We obtain successively

k	x^k	y^k
0	0.2	1.3
1	0.11	1.77
2	-0.056	1.680
3	-0.002	1.682
4	-0.0094	1.6872
5	-0.0086	1.6849
6	-0.0085	1.6853

Zaguskin (1961).

§ 3.33 Gauss-Seidel Iteration

As in the method of simple iteration we construct a sequence of vectors by the iterative equation:

$$x_i^{k+1} = f_i(x_1^{k+1}, \dots, x_{i-1}^{k+1}, x_i^k, \dots, x_n^k) , \quad i=1,2,\dots,n,$$

where the notation implies that the $(k+1)$ iterative components are used as they become available. Speed of convergence is often faster for Gauss-Seidel iteration than for simple iteration, Zaguskin (1961).

§ 3.4 Gradient Methods

The most important methods available for solving the nonlinear system are the "gradient methods" of minimization. These methods use measurements of the slope of the function as indicators of the direction toward the desired solution. The successive approximations to the solution are

determined by the iterative equation:

$x^{k+1} = x^k + t_k \Delta x^k$, where t_k determines the step-size and Δx^k the direction. The various gradient methods (cf. § 2.65) are dependent upon the particular choice of t_k and Δx^k , Spang (1962). The first approximation is arbitrary but should be picked as close as possible to the true solution. In practice, the slope of the function can usually be approximated numerically. The most recent computational experience using gradient and conjugate gradient methods are available in Rosenbrock (1960), Martin and Tee (1961), and Powell (1962).

A recent trend in gradient methods stresses simplicity of coding for a stored program computer; in particular, the calculation of derivatives tends to be avoided. The extremes of these almost purely experimental "downhill" methods are available in Ward (1957) and Lance (1959) and are reviewed in this chapter.

§ 3.41 Univariate or Relaxation Methods

The simplest and crudest gradient method is obtained by varying only one coordinate component per iterative step. The particular coordinate can be chosen in a cyclic or any arbitrary manner. One method is to use the coordinate defined by

$$\max_i \left| \partial T(x^k) / \partial x_i \right|, \quad i = 1, 2, \dots, n.$$

A slight modification of the above method is the Southwell-Synge method, discussed in Booth (1949), where the coordinate x_j is chosen by

$$\left| (\partial T / \partial x_j)^2 / 2 \partial^2 T / \partial x_j^2 \right| = \max_i \left| (\frac{\partial T}{\partial x_i})^2 / 2 \frac{\partial^2 T}{\partial x_j^2} \right|, \quad i=1,2,\dots,n.$$

In both cases the step-size t_k can be chosen from the truncated Taylor series for $T(x^{k+1})$ about the point

x^k . Thus,

$$t_k = \frac{\partial T(x^k)}{\partial x_j} / \frac{\partial^2 T(x^k)}{\partial x_j^2} .$$

We note here that the second derivative must be positive.

§ 3.42 Newton's Method

The extension of Newton's Method to the non-linear system $f(x) = 0$; $f(x)$, $x : nx1$ gives the iterative process:

$$x^{k+1} = x^k - J_k^{-1} f(x^k) , \text{ where } J_k = \left[\frac{\partial f_i(x^k)}{\partial x_j} \right] .$$

Newton's method for the nonlinear system is usually written in the form:

$$J_k(x^{k+1} - x^k) = - f(x^k) .$$

The problem of minimizing the test function $T(x)$ can be written as an iterative equation in a form appropriate to Newton's method as:

$$H(x^{k+1} - x^k) = -\nabla T , \text{ where } H = \left[\frac{\partial^2 T(x^k)}{\partial x_i \partial x_j} \right] .$$

Convergence in the large, i.e., convergence to a solution from an arbitrary starting point, is not assured. A starting point symmetrically placed between two solutions will oscillate between the two solutions and never converge if no rounding error is introduced. To prevent initial overshoot - a common occurrence - step limitation is often desirable. A nonconvergent sequence can sometimes be overcome with a restart at a different initial approximation.

Experience suggests that Newton's method should be considered only for problems of limited scope, Turner (1960). A discussion of Newton's method is also available in Haselgrove (1961).

§ 3.42.01 First Modification of Newton's Method

If we replace $J_k = J(x^k)$ by $J_0 = J(x^0)$ in Newton's method we obtain the following:

$$x^{k+1} = x^k - J_0^{-1}f(x^k) .$$

This iteration is computationally easier per step but the speed of convergence is decreased. It can be shown that, if the initial approximation is sufficiently close to the exact solution, such a procedure will converge for a large class of functions $f(x) = 0$. This modification gives satisfactory results for problems in one or two variables but the extension to more variables will probably prove to be unsatisfactory.

§ 4.42.02 Second Modification of Newton's Method

This method is essentially equivalent to the first modification of Newton's method but differs in that a sequence of correction vectors X^1, X^2, \dots are constructed instead of the previous sequence of approximate solution vectors x^1, x^2, \dots .

The iteration equation is given by:

$$f(x^0 + X) \equiv f(x^0) + J_0 X + g(X) = 0 ,$$

$f(x)$; $X, g(X) : nx1$, where $g(X)$ is defined by the above identity. We may rewrite our iterative procedure

as:

$$X = Y + G(X) , \text{ where}$$

$$Y = -J_0^{-1} f(x^0) , \quad G(X) = -J_0^{-1} g(X) ,$$

$$J_0 = \left[\frac{\partial f_i(x^0)}{\partial x_j} \right] .$$

We then find successive correction vectors X^1, X^2 , by means of the formulas:

$$X^1 = Y$$

$$X^{k+1} = Y + G(X^k) .$$

Beckenbach (1956). An example illustrating the method is given in § 3.42.04.

§ 3.42.03 Example of Newton's Method

Let the nonlinear system and initial approximation be given as

$$x = \begin{pmatrix} y \\ z \end{pmatrix} , \quad f(x) = \begin{pmatrix} f(y,z) \\ g(y,z) \end{pmatrix} , \quad x^0 = \begin{pmatrix} 0.4 \\ 1.0 \end{pmatrix} ,$$

where $f(y,z) = y^3 + 2y^2 - 3yz^2 + 6yz - y - 2z^2 + z + 4$,

$$g(y,z) = -z^3 + 3z^2 + 3y^2z + 4yz - z - 3y^2 - y - 2 .$$

Now

$$J = \begin{bmatrix} \frac{\partial f}{\partial y} & \frac{\partial f}{\partial z} \\ \frac{\partial g}{\partial y} & \frac{\partial g}{\partial z} \end{bmatrix} = \begin{bmatrix} 3y^2 + 4y - 3z^2 + 6z - 1, & -6yz + 6y - 4z + 1 \\ 6yz + 4z - 6y - 1, & -3z^2 + 6z + 3y^2 + 4y - 1 \end{bmatrix} ,$$

$$J_0 = \begin{bmatrix} -7.920 & 9.800 \\ -9.800 & -7.920 \end{bmatrix} , \quad \begin{pmatrix} f(x^0) \\ g(x^0) \end{pmatrix} = \begin{pmatrix} -2.616 \\ 0.040 \end{pmatrix} .$$

Our linear system is

$$-7.920 \Delta y + 9.800 \Delta z = 2.616$$

$-9.800 \Delta y - 7.920 \Delta z = -0.040$, which has the solution $\Delta y = -0.12803$, $\Delta z = 0.16347$. The second approximation, using the iterative equation

$$x^1 = x^0 + \Delta x^0 , \text{ becomes}$$

$$x^1 = \begin{pmatrix} 0.27197 \\ -0.83653 \end{pmatrix} .$$

Repeating the process we obtain

$$x^2 = \begin{pmatrix} 0.24823 \\ -0.82096 \end{pmatrix} .$$

Exponential extrapolation of x^0 , x^1 , x^2 gives the result

$$x^3 = \begin{pmatrix} 0.24283 \\ -0.81933 \end{pmatrix} , \text{ which has an error } < 4 \times 10^{-3} .$$

§ 3.42.04 Example of Second Modification of Newton's Method

Let $f(x)$ and x^0 be given as

$$f(x) = \begin{pmatrix} f(y,z) \\ g(y,z) \end{pmatrix} = \begin{pmatrix} 4+y+z-y^2+2yz+3z^2 \\ 1+2y-3z+y^2+yz-2z^2 \end{pmatrix} = 0, x^0 = \begin{pmatrix} y^0 \\ z^0 \end{pmatrix} = \begin{pmatrix} 3.3 \\ -3.0 \end{pmatrix} .$$

$$\text{Then } J_0 = \begin{bmatrix} 1-2y+2z & , & 1+2y+6z \\ 2+2y+z & , & -3+y-4z \end{bmatrix}_{x_0} = \begin{bmatrix} -11.6 & -10.4 \\ 5.6 & 12.3 \end{bmatrix} ;$$

$$J_0^{-1} = \begin{bmatrix} -0.14566 & -0.12316 \\ 0.06632 & 0.13738 \end{bmatrix} , f(x^0) = \begin{pmatrix} 0.61 \\ -0.41 \end{pmatrix} .$$

$$\text{Now } Y = -J_0^{-1} f(x^0) = \begin{pmatrix} 0.0384 \\ 0.0159 \end{pmatrix} = x^1 .$$

The successive correction vectors X^1 , X^2 are now obtained

$$\begin{aligned} X^1 &= Y = \begin{pmatrix} 0.0384 \\ 0.0159 \end{pmatrix} \\ G(X^1) &= J_0^{-1}g(X^1) = -J_0^{-1}f(x_0+X^1) = \begin{pmatrix} 0.000227 \\ -0.000281 \end{pmatrix} \\ X^2 &= Y + G(X^1) = \begin{pmatrix} 0.038627 \\ -0.015419 \end{pmatrix} \\ G(X^2) &= J_0^{-1}f(x_0+X^2) = \begin{pmatrix} -0.000005 \\ 0.000000 \end{pmatrix} \\ X^3 &= Y + G(X^2) = \begin{pmatrix} 0.038622 \\ -0.015419 \end{pmatrix}. \end{aligned}$$

Hence our final answer correct to 5 decimal places is given by $\alpha = x_0 + X^3$, where

$$\alpha = \begin{pmatrix} 3.33862 \\ -2.98438 \end{pmatrix}.$$

An important advantage of the second modification of Newton's method is that we can increase the number of significant figures in successive approximations by computing successively diminishing error vectors.

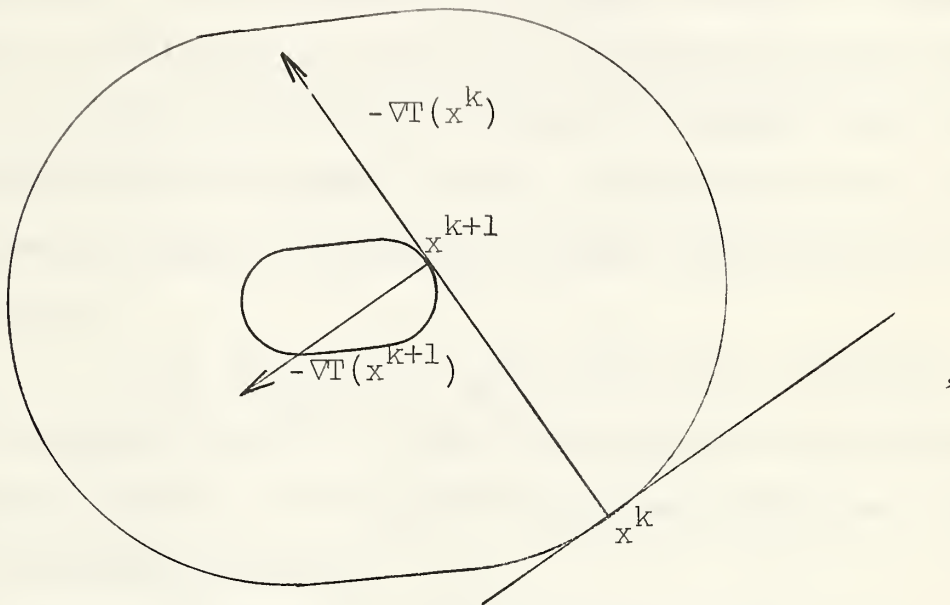
§ 3.43 Methods of Descent

Let us consider the test function $T(x) = (f(x))^T f(x)$ in connection with the nonlinear system $f(x) = 0; f(x), x: n \times 1$. Clearly $T(x)$ will have a solution, if, and only if $x = \alpha$, where α is a solution of $f(x) = 0$. From elementary vector analysis we recall that the direction of maximum decrease in the value of $T(x)$ is given by the negative gradient $-\nabla T(x)$.

Let $x = x(t)$, where $x(0) = x^0$ and $t > 0$, define a curve in n -space. We now define the test function $T(x)$ as $G(t)$ where every permissible t defines a corresponding point on the curve $x = x(t)$ and a value for $T(x)$. If we follow the curve $x = x(t)$, $t \geq 0$ defined by $\frac{dx}{dt} = -\nabla T(x)$, $x(0) = x^0$, we will at least momentarily follow the path of steepest descent and in general we can find a value of $t > 0$ such that

$$G(t) = T(x^0 - t \nabla T(x^0)) < T(x^0).$$

In this manner we construct a sequence of points x^0, x^1, \dots such that $T(x^{k+1}) < T(x^k)$. Under suitable conditions we will have convergence to a solution of $T(x)$. The value of t required at each step is determined either by trial and error or by finding the smallest positive root of $G(t)=0$. This variation of the method has the geometrical interpretation illustrated by the following diagram



which represents the section of surfaces $T(x^k) = c_k$, $T(x^{k+1}) = c_{k+1}$ intersected by the plane through x^k containing the directions $-\nabla T(x^k)$ and $-\nabla T(x^{k+1})$. The surface we seek $T(x^{k+1}) = c_{k+1}$ must have $-\nabla T(x^k)$ as tangent and clearly the subsequent directions of steepest descent will be at right angles to each other; i.e., we have arrived at that point on the line of steepest descent where the derivative of $T(x)$ with respect to the distance along the line is zero.

The methods of "descent" have several advantages: they usually converge to a solution, they allow approximations to be used, and they can give more than one solution if several are required. The recent trend in descent methods is to stress simplicity of coding for a stored program computer. Ward (1957) and Lance (1959) have developed methods in which the level points are sought by systematic sampling techniques.

§ 3.43.01 Method of Steepest Descent

We now consider the Method of Steepest Descent in greater detail. Let the test function $T(x)$ be given as

$T(x) = (f(x))^T f(x)$; furthermore, let x^0 be an initial approximate solution. Numerical methods for determining the minimum of $T(x)$ can be easily devised by using the geometrical picture. In the neighbourhood of the solution $T(x)$ represents - by assumption - a convex surface. Starting with x^0 sufficiently close to the solution and proceeding in the proper direction Δx^0 along any straight line (except the one which is tangent to $T(x) = \text{constant}$) we can always arrive at a point x^1 which is a better approximation to the solution.

Let $x^{k+1} = x^k + t_k \Delta x^k$ define the iteration. If Δx^k is chosen by $\Delta x^k = - \nabla T(x^k)$ we have the direction of steepest descent. The optimum t_k is calculated as a root of the equation (in t):

$$\frac{\partial}{\partial t} T(x^k - t \nabla T(x^k)) = 0 .$$

In the calculation of t_k there is no object in striving after great accuracy. A relative error of several percent should not greatly affect the rapidity of convergence. The solution of the above optimization problem may be effected in a satisfactory manner by determining the first point on the directed line segment - defined by an initial point x^k and direction $-\nabla T(x^k)$ - for which the direction of steepest descent at that point is perpendicular to the given line. More explicitly: let the directed line segment be defined by x^k , $s^k = -\nabla T(x^k) / | \nabla T(x^k) |$; the point on the line is given as $x^{k+1} = x^k + t_k s^k$; and the condition to be satisfied is $(s^{k+1})^T s^k = 0$. The desired value of t_k can be determined

very satisfactorily by linear interpolation: the following approximate value determined by Newton's Method of t_k is also used if $x^k \approx \alpha$:

$$t_k \approx \frac{T(x^k)}{(\nabla T(x^k))^T \nabla T(x^k)} .$$

Again, we illustrate the above procedure by the following example.

§ 3.43.02 Example of Steepest Descent

Let $f(x) = 0$ be defined as

$$f_1(x, y, z) = x^2 + y^2 + z^2 - 9 = 0$$

$$f_2(x, y, z) = x + y + z - 3 = 0$$

$$f_3(x, y, z) = x - y = 0 .$$

Our test function $T(x) = (f(x))^T f(x)$ becomes

$$T(x, y, z) = (x^2 + y^2 + z^2 - 9)^2 + (x + y + z - 3)^2 + (x - y)^2 .$$

The direction of steepest descent is given by $-\nabla T(x, y, z)$,

$$\text{where } \nabla T(x, y, z) = \begin{pmatrix} 4x^3 + 4xy^2 + 4xz^2 - 32x + 2z - 6 \\ 4y^3 + 4yx^2 + 4yz^2 - 32y + 2z - 6 \\ 4z^3 + 4zx^2 + 4zy^2 - 34z + 2x + 2y - 6 \end{pmatrix} .$$

$$\text{Let } x^0 = (x^0, y^0, z^0) = (0, 0, 1) .$$

We shall use the approximation

$$t_k \approx \frac{T(x^k, y^k, z^k)}{(\nabla T(x^k, y^k, z^k))^T \nabla T(x^k, y^k, z^k)}$$

for the step-size problem.

STEP 1 $T(0,0,1) = (-8)^2 + (-2)^2 = 68$

$$\nabla T(0,0,1) = \begin{pmatrix} -4 \\ -4 \\ -36 \end{pmatrix}$$

$$t_0 = \frac{68}{(-4)^2 + (-4)^2 + (-36)^2} = \frac{68}{1328} \approx \frac{1}{20}.$$

Hence $x^1 = x^0 - t_0 \nabla T(x^0)$ gives

$$\begin{array}{rcccl} x^1 & x^0 & & & \\ (y^1) & (y^0) & - \frac{1}{20} \begin{pmatrix} -4 \\ -4 \end{pmatrix} & = & (0) + \left(\frac{1}{5}\right) = (0.2) \\ z^1 & z^0 & -36 & 1 & \frac{9}{5} = 2.8 \end{array}$$

STEP 2 $T(0.2,0.2,2.8) \approx 1.2$

$$\nabla T(0.2,0.2,2.8) = \begin{pmatrix} -.46 \\ -.46 \\ -11.70 \end{pmatrix}$$

$$t_1 \approx \frac{1.2}{(-12)^2} \approx \frac{1}{120}$$

Hence $x^2 = x^1 - t_1 \nabla T(x^1)$ gives

$$\begin{array}{rcccl} x^2 & 0.2 & & & \\ (y^2) & (0.2) & + \left(\frac{1}{120}\right) \begin{pmatrix} .46 \\ .46 \end{pmatrix} & \approx & (0.2) \\ z^2 & 2.8 & 11.70 & & 2.9 \end{array}$$

STEP 3 $T(0.2,0.2,2.9) = .25 + .09 = .34$

$$\nabla T(0.2,0.2,2.9) = \begin{pmatrix} 0.19 \\ 0.19 \\ -5.32 \end{pmatrix}$$

$$t_2 \approx .34/28.3 \approx 1/80.$$

Hence, $x^3 = x^2 - t_2 \nabla T(x^2)$ gives

$$\begin{array}{cccc} x^3 & 0.2 & 0.2 & 0.2 \\ (y^3) & = (0.2) & - 1/80 (0.2) & = (0.2) \end{array} .$$

$$\begin{array}{cccc} z^3 & 2.9 & -5.32 & 2.96 \end{array}$$

STEP 4 $T(0.2, 0.2, 2.96) \approx .15$

$$\nabla T(0.2, 0.2, 2.96) \approx \begin{pmatrix} .58 \\ .58 \\ -1.15 \end{pmatrix}$$

$$t_3 \approx .15/2.01 \approx .075$$

Hence, $x^4 = x^3 - t_3 \nabla T(x^3)$ becomes

$$\begin{array}{cccc} x^4 & 0.2 & .58 & .15 \\ (y^4) & = (0.2) & - .075 (.58) & = (.15) \\ z^4 & 2.96 & -1.15 & 2.97 \end{array} .$$

We observe that the rate of convergence in the region of the true solution (0,0,3) is extremely slow. This example illustrates the inherent weakness of the method. Hence, in the region of a stationary value other methods, e.g. Newton's method, is preferable to the method of Steepest Descent.

§ 3.44 Steepest Descent by Integration of a set of Ordinary Differential Equations

Let us at this point sketch in quickly an approach which follows naturally from the methodology of steepest descent. The nonlinear system $f(x) = 0$; $f(x)$, $x : nx1$, can be replaced by the initial value problem

$$\frac{dx}{dt} = - \nabla T(x) ; x(0) = x^0 .$$

The above system can be solved analytically only for very simple problems but numerical integration may prove to be satisfactory. The interval size and the number of iterations allowed in numerical methods before a restart is required will in general depend upon the particular problem, Bechenbach (1956). Despite its obvious interest this approach has not been much explored and its potential value remains unassessed.

§ 3.45 Conjugate Gradient Method of Steepest Descent

In many ways the conjugate gradient method is similar to the method of steepest descent previously reviewed in § 3.43.01; however, this recent method is quadratically convergent. A disadvantage of the method of steepest descent is its poor convergence near a stationary value. This difficulty can be overcome by switching to another method such as Newton's method which is quadratically convergent near a stationary value. Gradient methods which have second-order convergence for finding the minimum of a quadratic positive definite function have already been reviewed in § 2.65 and are extensively considered in Martin and Tee (1952). A method - in some ways similar to the conjugate gradient method of Hestenes and Stiefel (1952) - which finds stationary values of a general function in several variables is given by Powell (1962). This particular method which has second-order convergence is worth outlining in some detail.

§ 3.45.01 Basic Assumption of the Method

From the general nonlinear problem

$$f(x) = 0 ; f(x), x : n \times 1 ,$$

we construct the non-negative test function

$$T(x) = (f(x))^T f(x) .$$

Let $G(x)$ be an approximation to $T(x)$ in the neighbourhood of the stationary value α , i.e.,

$$G(x) = T(\alpha) + \frac{1}{2}(x-\alpha)^T H(x-\alpha) ,$$

where $H = [\partial^2 T(\alpha) / \partial x_i \partial x_j]$. Then as a consequence of the definitions and assumptions,

$$\frac{\partial T}{\partial x_i}(\alpha) = 0 \quad i = 1, 2, \dots, n,$$

and $\partial T^2(\alpha) / \partial x_i \partial x_j = \partial T^2(\alpha) / \partial x_j \partial x_i$.

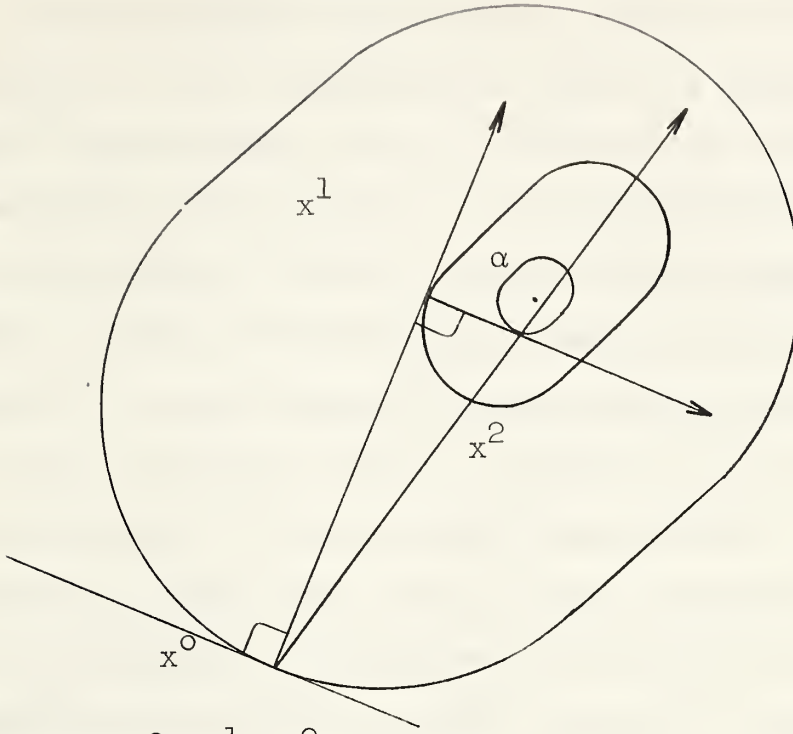
The iterative procedure will have second-order convergence if and only if the iteration leads from an arbitrary estimate of the stationary value, say x^0 , to α in a single cycle, i.e. when

$$T(x) \equiv G(x) \text{ .}$$

The method is based on the corollary of the theorem that if $G(x)$ is quadratic in the independent variables then any line which passes through α intersects the members of the family of contours $G(x) = C$ at equal angles. The corollary is that if the normal at t ; $t : n_{x1}$, to the contour $G(x) = G(t)$ is parallel to the normal at t^1 ; $t^1 : n_{x1}$, to $G(x) = G(t^1)$ then the line joining t to t^1 passes through α . Before we proceed to define the procedure in the general case we will discuss the two-dimensional case.

§ 3.45.02 Conjugate Gradient in Two Dimensions

The manner in which the iteration proceeds for the two-dimensional case can be most easily described with the assistance of the following diagram:



where x^0 , x^1 , x^2 , $\alpha : 2x^1$.

In the above figure, x^0 is the initial estimate and x^1 is a point on the line defined by $-\nabla T(x^0)$ and x^0 . The point x^1 may be any point on the line which is a finite distance away from x^0 , however, if it is chosen as the point where the derivative of $T(x)$ with respect to the distance along the line is zero then the convergence of the process is assured. The point x^2 is now obtained in a similar manner subject to the constraint that $(x^2 - x^1)^T (x^1 - x^0) = 0$; i.e, the successive directions are conjugate to each other. Clearly, α lies on the line determined by x^2 and x^0 .

If, as is usual, $T(x) \neq G(x)$, the same recipe is used for each cycle, and, in general, α will not be found by a single iteration.

§ 3.45.03 Conjugate Gradient in n-Dimensions

The previous description of an iterative cycle in two-dimensions allows us to formulate the general procedure for the n-dimensional case in a simple manner. In the n-dimensional case the procedure now consists of evaluating the gradient at x^0 and determining a point x^1 , on the gradient vector, at which the derivative of $T(x)$ with respect to the distance along the line is zero. Of course, we may choose x^1 as any point on the line which is a finite distance from x^0 ; however, for convergence to be assured, we choose a good approximation to the optimum distance. The second step of each iterative cycle consists of determining a stationary value $T(x)$ in the $(n-1)$ - dimensional hyperplane which contains x^1 and is defined by the normal line $x^0 - x^1$. Let this stationary value be denoted as x^{2n-2} . Since the normal to $T(x) = T(x^{2n-2})$ at $x = x^{2n-2}$ is parallel to the normal to $T(x) = T(x^0)$ at $x = x^0$ the required stationary value in n-space will be that point on the line joining x^0 to x^{2n-2} where the derivative of $T(x)$ with respect to the distance along the line is zero. In the process described the steepest descent at a point has to be evaluated first in n-dimensions, then in $(n-1)$ - dimensions, ..., and finally in one dimension. Also, on $(2n-1)$ occasions, knowing the derivatives at a point on a line we have to calculate the point where the derivative with respect to the distance along the line is zero. We recognize that the first $(n-1)$ times in a cycle this is done it is not essential, while it is essential to find

the stationary value the last n times. The numerical results available in Powell (1962) suggest that this conjugate gradient method is much more powerful than that of steepest descents; furthermore, since it can be programmed without too much difficulty, it is very suitable for electronic computing calculations.

§ 3.45.04 Example of the Conjugate Gradient Method

The following example defined as

$$T(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$$
$$x^0 = (3, -1, 0, 1)$$

was chosen in order that the conjugate gradient method could be compared with the work of previous authors.

Table I Comparison of Results

k	steepest descent	Booth's variation	conjugate gradient
0	215.000	215.000	215.000
7	6.355	5.352	0.009
14	3.743	0.620	9×10^{-5}
21	2.269	0.135	2×10^{-6}
28	1.420	0.051	2×10^{-6}
35	0.919	0.009	1×10^{-6}
42	0.614	0.008	5×10^{-8}
49	0.423	0.008	4×10^{-9}

We note that in four dimensions the conjugate gradient method requires a minimum to be calculated in

(2n-1) or 7 distinct directions in order to complete a cycle and hence the comparison is as above, Powell (1962).

The second iteration is as follows.

Table II The second Iteration of the Method

point	x_1	x_2	x_3	x_4	$T(x)$
A	0.1266	-0.0123	0.1455	0.1428	.00852
B	0.1263	-0.0104	0.1337	0.1441	.00699
C	0.1260	-0.0124	0.1333	0.1435	.00659
D	0.1259	-0.0111	0.1331	0.1391	.00632
E	0.1229	-0.0107	0.1332	0.1392	.00632
F	0.1229	-0.0107	0.1332	0.1392	.00632
G	0.1166	-0.0114	0.1323	0.1303	.00583
H	0.0396	-0.0044	0.0300	0.0332	.00009

The reason that the conjugate gradient method does not have second-order convergence in this case is that $T(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2$ near the minimum, hence it does not determine the minimum uniquely.

In the second iteration the first and second steps from A to B and B to C correspond to the ordinary steepest descent method; the steps from C to D and D to E correspond to steepest descent in two and one dimensions, respectively. Since E nearly coincides with D - and this is a common occurrence - the minimum on CE, F, nearly coincides with E. The finding of G, the minimum on BF, gives a slight improvement in $T(x)$, and the final step, finding H on AG,

reduces the value of $T(x)$ handsomely.

§ 3.45.05 Computational Considerations

The numerical computations in the construction of the conjugate gradient vectors can be effected in a satisfactory and efficient manner by using projection matrices - defined in § 5.21.07. We define Q as the intersection of q linearly independent hyperplanes in n -space and P_q as the projection matrix which takes any vector in n -space into the intersection Q . Let us define the successive hyperplanes required in the conjugate gradient method by $n_i^T x = b_i$, normalized so that $n_i^T n_i = 1$, $i = 1, 2, \dots, q$. We have, therefore,

$$Q : N_q^T x = b, \quad N_q \equiv [n_1, n_2, \dots, n_q] : nxq,$$

and $P_q : I_n - N_q (N_q^T N_q)^{-1} N_q^T$, where $I_n : nxn$ unit matrix. Since the successive hyperplanes are conjugate to each other the successive projection matrices can be obtained recursively from

$$P_q = P_{q-1} - n_q n_q^T, \quad P_0 = I_n.$$

At the q -th stage of each iterative cycle we require both the direction of steepest descent in $(n-q)$ - dimensions and the optimum step-size in that direction. Clearly, the direction of steepest descent under these constraints is given by

$$d^{q+1} = P_q \nabla T(x^q), \quad q=0, 1, \dots, n-1, \quad P_0 = I_n.$$

A suitable numerical method of obtaining the optimum step-size along d^{q+1} is the method of linear interpolation or the method of bisections. Computing experience

indicates, however, that the point obtained by linear interpolation be subsequently checked for accuracy. The following computing algorithm describes a method which has been found to be successful.

Given : $(P_k; t_k; x^k)$; Find : $(P_{k+1}; t_{k+1}; x^{k+1})$;

where $P_k, P_{k+1} : nxn$; $x^k, x^{k+1} : nx1$; $t_k, t_{k+1} : \text{scalars}$.

	<u>conditional branch</u>	<u>comments</u>
73 ; $l \leftarrow t_k$		$t_k = x^k - x^{k-1} $
$t_2 \leftarrow 1.0$		
$t \leftarrow 0.0$		
$xtemp \leftarrow x^k$		
9408 ; $h_1 \leftarrow t$		
$t \leftarrow t+l$		$t \approx x^{k+1} - x^k $
$t_1 \leftarrow t_2$		
086 ; $x \leftarrow xtemp - txs$		$x \approx x^{k+1}, s = g(x^k) / g(x^k) $
$g \leftarrow -\nabla T(x)$		
$stemp \leftarrow g / g $		
71 ; $t_2 \leftarrow s^t stemp$		
$ t_2 : \epsilon$	$, \xrightarrow{\leq} 4540$	is $ g(x)^T g(x^k) \approx 0$?
$t_2 : 0$	$, \xrightarrow{\geq} 9408$	is t too large?
$t \leftarrow t+l(t_2/t_1 - t_2)$		linear interpolation for new t
$l \leftarrow h_2 - h_1$		


```

                                → 086
4540 ; tk+1 ← t                                tk+1 = | xk+1 - xk |
        xk+1 ← xk + tk+1 s
        Pk+1 ← Pk - sst                                another constraint added
                                END

```

In §3.45.06 we incorporate the above in the basic computing algorithm.

At the end of each stage for the first (n-1) stages we add another constraint or hyperplane to our projection matrix; at the end of the (n-1)-st stage we have the problem of determining the direction of steepest descent in n-space which lies on (n-1) hyperplanes. The following sequence is then available : x^0, x^1, \dots, x^n .

We now have the problem of calculating the stationary value x^{n+1} in the direction $x^{n-2} - x^n$, x^{n+2} in the direction $x^{n-3} - x^{n+1}$, ..., x^{2n-1} in the direction $x^0 - x^{2n-2}$. x^{2n-1} is the final value in each iterative cycle. The optimum step-size can again be determined by some method such as bisection or linear interpolation.

§ 3.45.06 The Basic Computing Algorithm

The following computing algorithm has been tested on the 1620 IBM electronic computer for the nonlinear system defined by:

$$r(x) = \begin{pmatrix} x_1 + 10x_2 \\ \sqrt{5}(x_3 - x_4) \\ x_2 - 2x_3 \\ \sqrt{10}(x_1 - x_4) \end{pmatrix} ; x^0 = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 1 \end{pmatrix} ; n=4, h=1.0, e=0.1, e1=0.001;$$

$P, I_n: nxn; xm: nx(n+1); g, s, stemp, x, xtemp, f: n \times 1.$

		comments
5408;	$P \leftarrow I_n$	$P_0 = I_4$
	$i \leftarrow 1$	
	$t \leftarrow 1.1$	value later req'd to end conjugation
	$h_2 \leftarrow h$	
	$l \leftarrow 2$	
7;	$g \leftarrow -\nabla T(x)$	$T(x) = (f(x))^T f(x)$
	$l : 2$	$\neq 9$ is g temporary
	$s \leftarrow Pg$	defines directed line
	$g \leftarrow s$	
	$t_2 \leftarrow g^T g$	
	$P \leftarrow P - ss^T / t_2$	defines new projec- tion matrix
9;	$t_2 \leftarrow g $	
	$stemp \leftarrow g / t_2$	
	$l : 2$	$\leq 71, \geq 73$
	$s \leftarrow stemp$	defines unit vector on directed line
	$xm(i) \leftarrow x$	x^1 stored for later use
73;	$h_3 \leftarrow h_2$	linear interpolation begins
	$t_2 \leftarrow 1.0$	see §3.45.05


```

        h2 ← 0.0
        xtemp ← x
9408;    h1 ← h2
        h2 ← h2+h3
        t1 ← t2
086;    x ← xtemp-h2xs
        l ← 1
        → 7
71;    t2 ← sTstemp
        | t2 | : e , ≤ 4540
        t2 : 0 , ≥ 9408
        h2 ← h2+h3( $\frac{t_2}{t_1 t_2}$ )
        h3 ← h2-h1
        → 086
4540;    i ← i+1                                next case
        fvalue ← f(x)
        n : 1 , ≤ 4401                          is the conjugation phase
        l ← 2                                    completed?
        → 7
4401;    t ← t-1.0                                stop conjugation process
        t : 0 , ≤ 632
        l1 ← l1-1
        e ← 0.2e                                increase accuracy req'd in
                                                step-size problem
632;    l1 ← l1-1
        l1 : 0 , = 4540                          is iterative cycle complete?
        xm(i) ← x                                store previous x

```


$s \leftarrow xm(\ell_1) - xm(i)$	determine direction of search
$s \leftarrow s / s $	
$x \leftarrow xm(\ell_1)$	define starting point of search
$i \leftarrow i-1$	
$\ell : 3$	
$\rightarrow 7$	
4541; fvalue : e_1	, \rightarrow END
$xm(1) \leftarrow x$	take last result as new initial value
$h \leftarrow 0.2h$	define new initial step-size
$e \leftarrow 5.0e$	
$\rightarrow 5408$	branch to restart
END	

§ 3.46 A Mixed Method

Davidon (1959) has described methods for locating a minimum of a quadratic test function, involving the calculation of an approximate inverse of the Hessian $[\partial^2 T / \partial x_i \partial x_j]$ without requiring that the Hessian be computed explicitly.

Starting with x^0 as an initial approximation to α and H^0 as an approximation of the inverse H^{-1} , Davidon computed the gradient $\nabla T(x^0)$ and a new test point \bar{x}^0 , $\bar{x}^0 = x^0 - H^0 \nabla T(x^0)$. If $T(\bar{x}^0) < T(x^0)$ or if there is a minimum of T between x^0 and \bar{x}^0 , the approximate inverse is modified so that the smallest value of T would have been arrived at in one step:

$$x^1 = x^0 - H^1 \nabla T(x^0).$$

The difference matrix $H^1 - H^0$ is so chosen that the step $x^1 - x^0$ is multiplied by a proper scalar so that all perpendicular steps would be unchanged. The process is repeated until the predicted change is less than some specified limit.

The method of modification depends upon the following results from matrix algebra, v : column vector, v^t its transpose, and b a scalar, then the matrix

$$H^1 = H^0 + b \frac{(H^0 v)(H^0 v)^T}{v^T H^0 v}$$

has the property that

$$H^1 v = (1+b) H^0 v$$

and, if y is any vector such that

$$(H^0 v)^T v = 0, \text{ then}$$

$$H^1 y = H^0 y .$$

Further, if

$$\nabla T(x^0) = |H^0| , \text{ and}$$

$$\nabla T(x^1) = |H^1| , \text{ then}$$

$$\nabla T(x^1) = (1+b) \nabla T(x^0) .$$

Finally, if H^0 is singular and annuls a vector Z , then H^1 also annuls Z .

Davidon also describes a method which is equivalent to a conjugate gradient method when T is a quadratic form in the independent variables. Starting with x^0 , $\nabla T(x^0)$,

and H^0 , we find H^1 by:

$$\begin{aligned}\bar{x}^0 &= x^0 - H^0 \nabla T(x^0) , \\ x^1 &= \bar{x}^0 - H^1 \nabla T(\bar{x}^0), \text{ so that} \\ x^1 &= x^0 - H^1 \nabla T(x^0) .\end{aligned}$$

This is achieved by the adjustment

$$H^1 = H^0 - b(H^0 \nabla T(\bar{x}^0))(H^0 \nabla T(\bar{x}^0))^T$$

where $b = 1 / (\nabla T(\bar{x}^0))^T H^0 (\nabla T(\bar{x}^0) - \nabla T(x^0))$

if H^0 is symmetrical and if b can be computed. A more comprehensive treatment of the method can be found in Davidon (1959). This method has not been adequately tested beyond Davidon's comment that it appeared to be faster and more effective than other known computing methods.

§ 3.5 Direct Search Methods

The recent trend in descent methods is to stress simplicity in coding for a stored program computer; in particular, the calculation of derivatives tends to be avoided. The extremes of this trend occur in the downhill methods of Ward (1957) and Lance (1959), in which the zeros of test functions are sought by systematic sampling procedures. The phrase "direct search" is used to describe sequential examination of trial solutions involving comparison of each trial solution with the "best" obtained up to that time together with a strategy for determining what the next trial solution will be. The strategy used in selecting a new point is dictated by various aspects of the problem, including one's knowledge of the solution space, the rules of transition between states, and the rules for selecting trial points.

§ 3.51 Downhill Method of Ward and Lance

The test function preferred by both Ward and Lance is the absolute-value norm or Gerschgorin vector norm

$$T(x) = \sum_{i=1}^n |f_i(x)| ; x : nx1 .$$

By Ward's method, values of the test function are computed at an arbitrary starting point x^0 and at $x^0 \pm h e_i$, for $i = 1, 2, \dots, n$, where h is an arbitrary step-size and e_i the unit vector whose only non-zero component is in the i -th position. If each independent variable is changed by $\pm h$ provided that

$$T(x^0 \pm h e_i) < T(x^0)$$

and otherwise is left unchanged, a point \bar{x}^1 is reached which satisfies

$$T(\bar{x}^1) < T(x^0).$$

A "pattern move" is next made defining a point $x = x^1$ which consists of changing all independent variables by an amount equal to the sum of the changes made in those variables during the previous n moves or searches. The process is now repeated with the previous point x^1 as the starting point.

The event

$$T(x^k) > T(\bar{x}^k)$$

at the conclusion of the k -th iterative cycle results in x^k being ignored and replaced by \bar{x}^k . Eventually we arrive at a point \bar{x}^{k+1} which satisfies

$$T(\bar{x}^{k+1}) = T(x^k) ;$$

at this time we replace h by $h/2$ and repeat the previous calculations. This process of searches and pattern moves is continued until the step-size is reduced to a predetermined minimum.

Lance (1959) has used a variant of this procedure to approximate the gradient of the test function $T(x)$ from

the relations

$$\frac{\partial T}{\partial x_1} \approx - \frac{T(x^0) - T(x^0 + h e_1)}{h} , \text{ or}$$

$$\frac{\partial T}{\partial x_1} \approx \frac{1}{2} \left(\frac{T(x^0 - h e_1) - T(x^0 + h e_1)}{h} \right) ,$$

which lead, respectively, to the following approximations for x^1 :

$$x_i^1 = x_i^0 + T(x^0) - T(x^0 + h e_i) , \text{ or}$$

$$x_i^1 = x_i^0 + \frac{1}{2} (T(x^0 - h e_i) - T(x^0 + h e_i)) ,$$

$$i = 1, 2, \dots, n.$$

The first approach is preferred because fewer values of the test functions are required. Using the Gerschgorin vector norm sometimes leads to stalemate situations but this difficulty can be detected and remedied. Ward's method degenerates into a stalemate situation when the test function contains a "trough" which is inclined to the axis. This particular difficulty is treated effectively by Lance's variation. An example of a simple problem where Lance's method failed is given by

$$\begin{aligned} f_1(x_1, x_2) &= x_1 = 0 \\ f_2(x_1, x_2) &= x_2^2 = 0 \end{aligned} , \quad x^0 = \begin{pmatrix} x_1^0 \\ x_2^0 \end{pmatrix} = \begin{pmatrix} 0 \\ -.25 \end{pmatrix} .$$

In general, it can be said that, Lance's method may tend toward a stalemate situation when the test function has troughs along the coordinate axis.

To conclude this discussion of downhill methods the following properties may be noted:

(i) The descent or downhill methods seem to assure, in general, convergence in the large, but the rate of convergence and accuracy may be poor at multiple roots. The development of stalemate situations may prevent ultimate convergence.

(ii) A general technique can be devised for a wide variety of problems with little or no modification from problem to problem.

(iii) One important fault of these methods is that they will cause convergence to a point of indeterminacy of a system of equations, and generally will do so without giving warning of any kind (cf. the example above).

CHAPTER IV

LINEAR PROGRAMMING

§ 4.1 Introduction

The mathematical theory of Linear Programming is concerned with optimizing a linear expression subject to linear constraints.

Historically the general problem of linear programming was first developed and applied in 1947 by Dantzig, Wood, and their associates of the U.S. Department of the Air Force. At that time, this group was studying the feasibility of applying mathematics and related techniques to military programming and planning problems. The initial mathematical statement of the general problem of linear programming was made by Dantzig in 1947; he also devised a systematic procedure for solving the problem known as the Simplex method. Prior to this a number of problems (some unsolved) had been posed which required the optimization of a linear function subject to linear constraints. The more important examples include the transportation problem posed initially by Hitchcock (1941) and later by Koopmans (1949) and the diet problem of Stigler (1945). The first successful application of a high-speed electronic computer to the linear programming problem occurred in January, 1952, on the N.B.S. SEAC machine.

The fundamental literature in this field can be found in the published papers of the Cowles Commission for Research in Economics resulting from a conference held in

Chicago in June, 1949. Most of the conference papers are available in Koopmans (1951). In particular, the papers by Dantzig (1951a,b) can be said to define the new discipline of Linear Programming.

An important reference on the theory of linear inequalities is Kuhn and Tucker (1956). This reference includes papers which present a detailed exposition of the fundamental mathematical results which form a basis for the linear programming models: papers which answer purely mathematical questions that have appeared in the elaboration of the economic theory; papers which include explicit results such as the duality theorem of linear programming for independent, mathematical purposes; and finally, papers which consider problems on the economic application of the models.

Problems arise in a large number of fields to which the discipline of Linear Programming can be successfully applied and include, among others, the following: industrial applications, transportation problems, contract awards, military applications, marketing analysis, and production scheduling. The methods now available in Linear Programming are many and varied but for the purposes of this thesis we review only the basic Simplex and Dual Simplex methods. In this chapter is also included a definition of the linear programming problem, the fundamental theorems involved, and certain definitions required.

The references consulted in this review include Gass (1958), Kuhn and Tucker (1956), and Vajda (1961). The

The first reference Gass (1958) is an excellent introductory text to the new discipline of Linear Programming.

§ 4.2 Definition of the Problem

Linear Programming is concerned with optimizing a linear expression - the objective function - subject to linear constraints.

e.g. maximize* $p^T x = z$; $p, x : n \times 1$,

subject to $Ax \leq b$, $x \geq 0$; $A : m \times n$; $b : m \times 1$.

The constraints can be rewritten as

$$Ax + I_m y = b ; I_m : m \times m ; y : m \times 1 ,$$

where the y_i 's are called "slack variables".

The following compact notation is also used to define the problem.

maximize $\left\{ p^T x \mid Ax \leq b, x \geq 0 \right\}$, or

maximize $\left\{ p^T x \mid Ax + I_m y = b , x, y \geq 0 \right\}$;

$A : m \times n$, $I_m : m \times m$; $p, x : n \times 1$; $y, b : m \times 1$.

§ 4.3 Definitions and Theorems

§ 4.31 Definitions

§ 4.31.01 A feasible solution is a vector $x : (n+m) \times 1$ which satisfies the equations $Bx = b$, where $B = [A, I_m]$;

* We use the word maximize here and in subsequent work as standing for l. u. b. or limes superior of $f(x)$ under the condition $x \in R$. In this particular example we have $f(x) = p^T x$ and R is defined by $Ax \leq b$, $x \geq 0$.

$A : m \times n$, $I_m : m \times m$, and the constraints $x \geq 0$.

§ 4.31.02 A basic feasible solution has no more than m positive x_j .

§ 4.31.03 A non-degenerate basic feasible solution has exactly m positive x_j .

§ 4.31.04 An optimum feasible solution optimizes the objective function subject to the constraints.

§ 4.31.05 If u_1, u_2 are two points in n -space then a convex combination of u_1, u_2 is a point u such that

$$u = \alpha_1 u_1 + \alpha_2 u_2 , \text{ where } 0 \leq \alpha_1 \leq 1 \text{ and } \alpha_1 + \alpha_2 = 1.$$

A set K of points is convex only if $u_1 \in K$ and $u_2 \in K$ implies that a convex combination of u_1, u_2 denoted by $u \in K$.

§ 4.31.06 A point $u_1 \in K$ is an extreme point (vertex) of K if u_1 cannot be expressed as a convex combination of any two other points of K .

§ 4.31.07 The convex hull $C(K)$ of any given set of points K is the set of all convex combinations of sets of points from K . $C(K)$ is the smallest convex set containing K ;
e.g. $C(K)$ where K consists of a finite number of points is a convex polyhedron.

§ 4.31.08 Two linear programming problems are called dual if they are associated in the following way :

$$\begin{aligned} &\text{maximize } \left\{ p^T x = z \mid Ax \leq b , x \geq 0 \right\} , \\ &\text{minimize } \left\{ b^T y = w \mid A^T y \geq p , y \geq 0 \right\} . \end{aligned}$$

Duality is a symmetrical relation and we refer to one problem as the primal and to the other as the dual.

§ 4.31.09 A starting point of a more detailed investigation concerning linear inequalities is the dual homogeneous system

$$Ax \geq 0, x \geq 0; A : m \times n; x : n \times 1, \text{ and}$$

$$A^T u \leq 0, u \geq 0; u : m \times 1.$$

§ 4.32 Theorems

§ 4.32.01 The Farkas Theorem

If for all vectors x satisfying a system of homogeneous linear inequalities $Ax \geq 0$ we have that $p^T x \geq 0$, then p is a non-negative linear combination of the rows of the matrix $A : p = A^T u, u \geq 0; A : m \times n; x, p : n \times 1, u : m \times 1$.

Geometrically, if all the points which are on those sides of all hyperplanes $H_i : a_i x = 0$ ($i=1, \dots, m$), a_i the i -th row of A , lie also on that side of $p^T x = 0$ then p lies within the cone spanned by the a_i . Farkas (1902), Zoutendijk (1960), Vajda (1961), and Kuhn and Tucker (1956).

Proof: Let $C = \left\{ y \mid \exists u, u \geq 0, y = A^T u \right\}$ be a convex polyhedral cone in n -space. We shall show that if $p \notin C$ then $\exists x$ such that $Ax \geq 0$ and $p^T x < 0$. This will prove the theorem. Hence assume $p \notin C$. Let $q : n \times 1$ be the projection of p on C and $r = q - p, r : n \times 1$. It follows:

(i) for all $y \in C, (q-p)^T (y-q) \geq 0$, since < 0 implies that $p \in C$,

(ii) $r^T q = 0$, since $\neq 0$ implies that $p \in C$.

the first of these is the fact that the number of cases of
 the disease is not proportional to the number of persons
 exposed to the disease. This is shown by the fact that
 the number of cases is not proportional to the number of
 persons exposed to the disease.

the second of these is the fact that the number of cases

is not proportional to the number of persons

exposed to the disease.

the third of these is the fact that the number of cases

is not proportional to the number of persons

exposed to the disease. This is shown by the fact that
 the number of cases is not proportional to the number of
 persons exposed to the disease. This is shown by the fact
 that the number of cases is not proportional to the number of
 persons exposed to the disease.

the fourth of these is the fact that the number of cases
 is not proportional to the number of persons
 exposed to the disease. This is shown by the fact that
 the number of cases is not proportional to the number of
 persons exposed to the disease.

the fifth of these is the fact that the number of cases
 is not proportional to the number of persons
 exposed to the disease. This is shown by the fact that
 the number of cases is not proportional to the number of
 persons exposed to the disease.

the sixth of these is the fact that the number of cases
 is not proportional to the number of persons
 exposed to the disease. This is shown by the fact that
 the number of cases is not proportional to the number of
 persons exposed to the disease.

the seventh of these is the fact that the number of cases

is not proportional to the number of persons

From (1) and (11) we obtain $r^T y \geq 0$ for all $y \in C$. It follows that for the point corresponding to a_i in n -space we have $a_i^T \in C$; hence $a_i^T r \geq 0$, so that $A^T r \geq 0$ holds. Since moreover $p^T r = q^T r - r^T r = -r^T r < 0$ we see that r is the vector looked for.

§ 4.32.02 Key Theorem

The two dual homogeneous systems $Ax \geq 0$ and $A^T u = 0, u \geq 0$; $A : m \times n$; $x : n \times 1$, $u : m \times 1$, always have a solution x^0, u^0 , satisfying $Ax^0 + u^0 > 0$ and $u_i^0 a_i^T x^0 = 0$ for all i , where a_i is the i -th row of A .

The essence of this theorem is that the convex hull of finitely many halflines is the intersection of finitely many halfspaces, Kuhn and Tucker (1956), Good (1959), Vajda (1961), and Zoutendijk (1960).

Proof: If $a_i^T x \leq 0$ for all x satisfying $Ax \geq 0$ then by Farkas' Theorem - $a_i^T = A^T u$ for some $u \geq 0$, so that $A^T u^i = 0$ would hold for some vector $u^i \geq 0$ with $u_i^i > 0$. Hence either $a_i^T x^i > 0$ for some x^i satisfying $Ax^i \geq 0$, or $A^T u^i = 0$ for some $u^i \geq 0, u_i^i > 0$. Let $u^i = 0$ in the former case and $x^i = 0$ in the latter so that $a_i^T x^i + u_i^i > 0$ will hold for all i . Let $x^0 = \sum_{i=1}^m x^i$, and $u^0 = \sum_{i=1}^m u^i$, then:

- (i) $Ax^0 \geq 0$ follows from $Ax^i \geq 0$ for all i ,
- (ii) $A^T u^0 = 0$ follows from $A^T u^i = 0$ for all i ,
- (iii) $u^0 \geq 0$ follows from $u^i \geq 0$ for all i ,
- (iv) $Ax^0 + u^0 > 0$ follows from $a_i^T x^k + u_i^k \geq 0$ for all k and > 0 for $k = i$,
- (v) $u_i^0 a_i^T x^0 = 0$ follows from $u_i^h a_i^T x^0 = 0$ for all h .

This proves the theorem.

§ 4.32.03 Inconsistency Theorem

The system of linear inequalities $Ax < 0$ is inconsistent if, and only if, $A^T u = 0$ for some $u \geq 0$ and $u_1 > 0$ for at least one i , hence if, and only if, one of the rows of the matrix A is linearly dependent with non-positive coefficients on the other rows of A .

§ 4.32.04 Theorem

The set of all feasible solutions to a linear programming problem constitute a convex set whose extreme points correspond to basic feasible solutions.

§ 4.32.05 Theorem

The objective function assumes its optimum value at an extreme point of R .

§ 4.32.06 Theorem

Every extreme point has m linearly independent vectors associated with it.

§ 3.32.07 Theorem

There are at least $\binom{n}{m}$ sets of linearly independent vectors for the case $Ax = b$, $x \geq 0$; $A : m \times n$; $x : n \times 1$, $b : m \times 1$.

§ 4.32.08 The Duality Theorem

If either the primal or the dual problem has a finite optimum solution, then the other problem has a finite optimum solution and the extremes of the linear function

are equal, i.e., $\min w = \max z$ (cf. § 4.31.08) .

If either problem has an unbounded optimum solution, then the other problem has no feasible solutions.

§ 4.4 The Simplex Method

The Simplex method of Dantzig is the fundamental method in the discipline of Linear Programming. By the Simplex procedure, we can, once any basic (extreme-point) feasible solution has been found, obtain an optimum feasible solution in a finite number of steps. The approach consists of obtaining new feasible solutions where our choice is such that the value of the objective function never decreases in the maximization and never increases in the minimization problems. Since only a finite number of extreme point solutions exist the optimum value of the objective function will eventually be found. The numerical computations performed in using the Simplex method are simple and straightforward. The degenerate case - the case when too many hyperplanes pass through an extreme point - will lead to cycling in the method but this problem can be easily dealt with.

§ 4.41 Formulation of the Simplex Method

We now proceed to define the Simplex Method by means of an example.

$$\begin{aligned} &\text{maximize } \left\{ p^T x \mid Ax \leq b, x \geq 0 \right\}, \text{ or} \\ &\text{maximize } \left\{ p^T x \mid Ax + I_m y = b, x, y \geq 0 \right\}, \text{ where} \end{aligned}$$

$$p = \begin{pmatrix} -1 \\ 1 \end{pmatrix}; A = \begin{pmatrix} -2 & 1 \\ 1 & -2 \\ 1 & 1 \end{pmatrix}; I_m = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}; y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}; b = \begin{pmatrix} 2 \\ 2 \\ 5 \end{pmatrix}.$$

The above problem can be represented more economically by the following tableau

x_1	x_2	b	
-2	1	2	y_1
1	-2	2	y_2
1	1	5	y_3
1	-1		z .

The tableau is interpreted as:

$$-2x_1 + x_2 + y_1 = 2$$

$$x_1 - 2x_2 + y_2 = 2$$

$$x_1 + x_2 + y_3 = 5, \text{ and}$$

$$x_1 - x_2 + z = z_0.$$

Our initial basic feasible solution

$$\begin{array}{rcl} x_1 & & 0 \\ x_2 & & 0 \\ (y_1) & = & (2) \text{ gives an objective function value} \\ y_2 & & 2 \\ y_3 & & 5 \end{array}$$

for $z_0 = 0$. Clearly, our basis at this point consists of (y_1, y_2, y_3) .

The Simplex method allows us to determine the coordinates of a new extreme point by an interchange of variables which is accomplished as follows:

(i) We choose a pivot in A as follows:

(a) the pivotal column is first determined by the algebraically smallest element in the objective function, i.e. in the bottom double lined row (in our example the pivotal column is determined by $x_2 = -1$);

(b) the pivotal row is next determined by the algebraically largest ratio of $a_{i,\text{piv.col.}}/b_i$, $i=1,2,\dots,m$, (in our example $i = 1$ since $1/2 > 1/5 > -2/2$).

At this point our pivot or pivotal element is a_{12} .

(ii) Divide all quantities in the pivotal row by pivotal element.

(iii) Divide all quantities in the pivotal column by the negative of the pivotal value and replace the pivot by $1/\text{pivot}$.

(iv) Replace all other quantities a_{ij} ($i \neq \text{piv.row.}, j \neq \text{piv.col.}$) by $a_{ij} + (a_{\text{piv.row.},j}) \times (a_{i,\text{piv.col.}}) / (a_{\text{piv.row.},\text{piv.col.}})$.

(v) Interchange the variables defined by the pivot (in our example we interchange x_2 and y_1).

(vi) Repeat from (i) until all coefficients in the bottom double lined objective function row are positive. (N.B. In the following example we have incorporated a slight variation of the above procedure. The variation is that we multiplied all quantities in the pivotal row by the pivot at the end of (ii) above.)

e.g. Interchange 1.

x_1	x_2	b
-2	<u>1</u>	2
1	-2	2
1	1	5
1	-1	

$\xleftarrow{(i)b}$
 y_1
 $\xrightarrow{\begin{matrix} (ii) , (iii) \\ (iv) \ \& \ (v) \end{matrix}}$
 y_2
 y_3
 z

x_1	y_1	b
-2	1	2
3	2	6
3	-1	3
-1	1	2

$1x_2$
 y_2
 y_3
 z

$\uparrow (i)a$

Interchange 2.

x_1	y_1	b			y_3	y_1	b	
-2	1	2	$1x_2$		2/3	1/3	4	$1x_2$
-3	2	6	y_2	$(ii) , (iii)$	1	1	9	y_2
<u>3</u>	-1	3	y_3	$\xleftarrow{-(i)b} (iv) \& (v)$	1	-1	3	$3x_1$
-1	1	2	z		1/3	2/3	3	z

$\uparrow (i)a$

The progress in basic feasible solution and objective value is given by the sequence

$$\begin{array}{lll}
 x_1=0 & x_1=0 & y_3=0 \\
 x_2=0 & y_1=0 & y_1=0 \\
 \left(\begin{array}{l} y_1=2 \\ y_2=2 \end{array} \right) , & \left(\begin{array}{l} x_2=2 \\ y_2=6 \end{array} \right) , & \left(\begin{array}{l} x_2=4 \\ y_2=9 \end{array} \right) . \\
 y_3=5 & y_3=3 & 3x_1=3 \\
 z_0=0 & z_0=2 & z_0=3
 \end{array}$$

Our final tableau is now interpreted as:

$$\begin{aligned}
 2y_3 + y_1 + 3x_2 &= 12 \\
 y_3 + y_1 + y_2 &= 9 \\
 y_3 - y_1 + 3x_1 &= 3 , \text{ and} \\
 y_3 + y_1 + 3z &= 9 .
 \end{aligned}$$

Clearly, we have arrived at the optimal solution since y_1 and y_3 cannot be decreased and still remain positive; hence, no interchange of variable can increase the value of the objective function. This particular example can be solved graphically without difficulty. The importance of the Simplex method becomes evident, however, in a problem which

incorporates a large number of constraints in a large number of variables.

A more comprehensive treatment of the Simplex method is available in Vajda (1961) and Gass (1958).

§ 4.42 The Dual Simplex Method

From the definition of the dual and the duality theorem we observe that to each tableau of the Simplex method there corresponds a dual tableau, and to an optimal tableau of the primal problem there corresponds an optimal tableau of the dual problem. To fix our ideas, assume we have the maximization problem defined by:

$$\text{maximize } \left\{ p^T x \mid Ax \leq b, x \geq 0 \right\},$$

described by the tableau

x_1	x_2	\dots	x_n	b	
a_{11}	a_{12}	\dots	a_{1n}	b_1	y_1
a_{21}	a_{22}			b_2	y_2
\dots	\dots				
a_{m1}	\dots		a_{mn}	b_m	y_m
$-p_1$	$-p_2$	\dots	$-p_n$		z

From the dual defined by

$$\text{minimize } \left\{ b^T u \mid A^T u \geq p, u \geq 0 \right\} \text{ and depicted by the tableau}$$

$-u_1$	\dots	$-u_m$	p	
a_{11}	\dots	a_{m1}	$-p_1$	v_1
			$-p_2$	v_2
\dots	\dots	\dots	\dots	\dots
a_{n1}		a_{nm}	$-p_n$	v_n
b_1	\dots	b_m		w

we can clearly represent both the primal and the dual by the following tableau

	x_1	x_2	x_n	b	
$-u_1$	A : $m \times n$			b_1	y_1
$-u_2$			
$-u_m$				b_m	y_m
p	$-p_1$	$-p_2$ $-p_n$		z
	v_1	v_2 v_n	w	

We observe that in the dual problem we can assume that the quantities in the double-lined row all have a positive sign but we don't insist upon feasibility of the variables (though still restricting ourselves to basic solutions). Whenever we have a negative basic variable, we transform the tableau. Since the primal and its dual can both be depicted by the same tableau we note that we can solve the dual problem, assuming a solution exists, by solving the primal problem. We observe that, up to the final tableau, we are dealing in the dual problem with solutions which are not feasible. We have overshot the target and we must trace our steps back to a feasible solution which is then optimal.

4.5 The Transportation Problem

The transportation problem - posed initially by Hitchcock (1941) and later independently by Koopmans (1947) - can be described, in its simplest form, as follows:

A homogeneous product is to be shipped in the amounts a_1, a_2, \dots, a_m , respectively, from each of m shipping

origins and received in amounts b_1, b_2, \dots, b_n , respectively, by each of n shipping destinations. The cost of shipping a unit amount from the i -th origin to the j -th destination is c_{ij} and is known for all combinations (i,j) . The problem is to determine the amounts x_{ij} shipped over routes (i,j) so as to minimize the total cost of transportation.

Clearly, we can assume that the total amount shipped is equal to the total amount received; i.e.,

$$\sum_i a_i = \sum_j b_j .$$

Furthermore, since a negative shipment is not meaningful we restrict each $x_{ij} \geq 0$.

The mathematical formulation of the transportation problem is given by:

$$\text{minimize} \quad \sum_i \sum_j c_{ij} x_{ij}$$

subject to the constraints

$$\sum_{j=1}^n x_{ij} = a_i \quad i = 1, 2, \dots, m$$

$$\sum_{i=1}^m x_{ij} = b_j \quad j = 1, 2, \dots, n$$

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$$

$$\text{and} \quad x_{ij} \geq 0 \quad \text{all } i, j .$$

A more formal definition of the transportation problem using the previous notation is formulated as:

$$\text{minimize } \left\{ \sum_i \sum_j c_{ij} x_{ij} \mid X=a, X^T=b, \sum_i a_i = \sum_j b_j; x_{ij}, a, b \geq 0 \right\} ;$$

$$C, X : m \times n ; a : n \times 1, b : m \times 1 .$$

Another formulation of the transportation problem is given by:

$$\text{minimize } \left\{ c^T X \mid AX = P_0, X \geq 0 \right\} ;$$

$$X, c : m \times n, P_0 : (m+n) \times 1, A : (m+n) \times mn .$$

For $m = 2$ and $n = 3$, we obtain the following 5 (i.e., $m+n$) equations in 6 (i.e., mn) unknowns:

$$\begin{aligned} x_{11} + x_{12} + x_{13} &= a_1 \\ x_{21} + x_{22} + x_{23} &= a_2 \\ x_{11} + x_{21} &= b_1 \\ x_{12} + x_{22} &= b_2 \\ x_{13} + x_{23} &= b_3 . \end{aligned}$$

Clearly, the first equation is redundant and does not need to be included in the system. Generalizing, we note that one equation can always be eliminated, and the transportation problem reduces to $m+n-1$ independent variables in mn equations.

We observe that the transportation problem involves generally a sparse matrix in its formulation. Hence, the various techniques now available take advantage of the properties associated with sparse matrices. A more extensive treatment of the transportation problem is available in Gass (1958), Vajda (1961), etc.

CHAPTER V

NONLINEAR PROGRAMMING

§ 5.1 Introduction and Definition of the Problem

§ 5.11 Introduction

The linear programming model is too restricted an approach for many mathematical programming problems. The nonlinear programming problem requires that a nonlinear function be optimized subject either to linear or nonlinear constraints. A capital distinction between the two types of problem may be emphasized here: the optimum solution of the linear programming problem lies at a vertex of the region determined by the linear constraints, i.e. if the solution exists; in the nonlinear problem the optimum solution need not be on the boundary.

A special case of nonlinear programming defined as quadratic programming - the problem of minimizing a convex quadratic function (or its dual), subject to linear constraints - is studied at some length in this thesis. A number of methods are now available which range from a direct extension of the Simplex method to the more sophisticated "Method of Feasible Directions" and "Gradient Projection" methods. The "Gradient Projection" by Rosen (1960) can be described by a simple computing algorithm and solved efficiently on an electronic computer.

We do not discuss the more difficult nonlinear programming problem defined as convex programming - the problem of minimizing a convex function (or maximizing a

concave function) in a convex region. For other techniques, an extensive bibliography on linear and nonlinear programming has been compiled by Riley and Gass (1958).

§ 5.12 Definition of the Nonlinear Programming Problem

§ 5.12.01 The nonlinear programming problem requires e.g. that a convex function $f(x)$ be minimized subject to the constraints $h(x) \geq 0$, $x \geq 0$; $h(x) : m \times 1$, $x : n \times 1$, where, in the general case, both $f(x)$ and $h(x)$ are nonlinear. This problem is formulated more compactly as:

$$\text{Minimize } \left\{ f(x) \mid h(x) \geq 0, x \geq 0 \right\}; h(x) : m \times 1, x : n \times 1.$$

We shall assume that $f(x)$ is differentiable with continuous partial derivatives.

§ 5.12.01 A typical quadratic programming problem may now be defined as:

$$\text{maximize } \left\{ p^T x - \frac{1}{2} x^T C x \mid Ax \geq b, x \geq 0 \right\};$$

$C : n \times n$ symmetric, non-negative definite, $A : m \times n$; p , $x : n \times 1$, $b : m \times 1$. The constraints $Ax \geq b$ are also denoted by

$$a_i x \geq b_i, i = 1, 2, \dots, m; a_i : 1 \times n.$$

§ 5.2 Definitions and Theorems

§ 5.21 Definitions and Notation

§ 5.21.01 A point $x^b \in R$ is called a feasible point where R is the closed convex region which consists of all points for

which $e(x) \geq 0$, where $e_i(x^b) = a_i x^b - b_i$; $x : n \times 1$, $e : m \times 1$, $a_i : 1 \times n$.

§ 5.21.02 A useful normalization is $n_i = (k a)_i^T$ where k is defined by $n_i^T n_i = 1$, $i=1, 2, \dots, m$. The constraints $e(x) = Ax - b \geq 0$ can then be written $e(x) = N_m^T x - b^1 \geq 0$ where

$$N_m \equiv [n_1, n_2, \dots, n_m] : n \times m.$$

§ 5.21.03 The $(n-1)$ - dimensional manifold defined by

$e_i(x) = 0$ is a hyperplane which is denoted by H_i , i.e.

$H_i : e_i(x) = n_i^T x - b_i^1 = 0$. We use the convention that n_i points "into" R .

§ 5.21.04 The unit vector $s : n \times 1$ with initial point $x^b \in R$ is termed a feasible direction if $\exists \lambda_b > 0$ such that for all λ , $0 \leq \lambda \leq \lambda_b$, $x = x^b + \lambda s \in R$.

§ 5.21.05 The nonlinear programming problem has a constrained global maximum if $g(x^0) = \nabla f(x^0) \neq 0$ for $x^0 \in R$. The nonlinear programming problem has an interior global maximum if $g(x^0) = 0$ for $x^0 \in R$.

§ 5.21.06 The intersection of q linearly independent hyperplanes in n -space is an $(n-q)$ - subspace which is defined as:

$$Q : e_i(x) = a_i x - b_i, \quad i = 1, 2, \dots, q.$$

The remaining space is denoted by \bar{Q} and since $Q \wedge \bar{Q} = 0$ then for $v \in Q$, $w \in \bar{Q}$ ($v, w : n \times 1$) we have $v^T w = 0$.

§ 5.21.07 A matrix $\bar{P}_q \equiv N_q (N_q^T N_q)^{-1} N_q^T$ is a projection matrix which takes any vector in n -space into \bar{Q} ; $\bar{P}_q : n \times n$, symmetric. A matrix $P_q = I - \bar{P}_q$ is a projection matrix which

takes any vector in n -space into the intersection Q .

§ 5.22 Theorems and Lemmas

The following Theorems and Lemmas are considered in greater detail in Rosen (1960).

§ 5.22.01 Lemma

Let $x^b \in Q$. A necessary and sufficient condition that the point $x = x^b + s$ be in R is that $N_q^T s \geq 0$.

Proof: Assume that for some i , $i=1, 2, \dots, q$, say $i=1$, $n_1^T s < 0$. Then since

$$e_1(x^b) = n_1^T x^b - b_1 = 0 , \text{ we have}$$

$$e_1(x) = n_1^T (x^b + s) - b_1 = n_1^T s < 0 .$$

Thus x violates the constraint $e_1(x) \geq 0$, and is therefore not in R . This proves the theorem.

§ 5.22.02 Theorem I

Let n_i , $i=1, 2, \dots, q$, be a set of q linearly independent vectors, and P_q the projection matrix. A necessary and sufficient condition that a nonzero vector y be linearly independent of the set n_i , $i=1, 2, \dots, q$, is that $P_q y \neq 0$.

§ 5.22.03 Theorem II

Let x^b be a boundary point of R which lies on exactly q , $1 \leq q \leq n$, hyperplanes, which are assumed to be linearly independent. Let the intersection of these hyperplanes be the manifold Q . Then the point x^b is a constrained global maximum of $f(x)$, if and only if $P_q g(x^b) = 0$, and

$$(N_q^T \ N_q)^{-1} N_q^T g(x^b) \leq 0 .$$

§ 5.22.03 Quadratic Dual Theorem

Let the primal and dual quadratic programming problems be given as :

$$\text{minimize } \left\{ f(x) = p^T x + \frac{1}{2} x^T C x \mid Ax \geq b , x \geq 0 \right\} ;$$

C : $n \times n$ symmetric, non-negative definite, A : $m \times n$; p, x : $n \times 1$, b : $m \times 1$, and

$$\text{maximize } \left\{ g(u, y) = b^T y - \frac{1}{2} u^T C u \mid A^T y - C u \leq p , y \geq 0 \right\} ;$$

u : $n \times 1$, y : $m \times 1$. Then the optimum values of $f(x)$ and $g(u, y)$ are equal.

The proof of this theorem is available in Dorn (1958) and Vajda (1961).

§ 5.3 Quadratic Programming

Let us now consider the special case of nonlinear programming known as quadratic programming: the problem of minimizing a convex quadratic function (or its dual), subject to linear constraints. Three methods applicable to this problem are considered in this chapter; they are

- (i) convex programming by extension of the Simplex method,
- (ii) method of Feasible Directions,
- (iii) Gradient Projection method.

The last two methods can be extended to the more general problem but the procedures, in general, are then no longer finite.

§ 5.31 Convex Quadratic Programming by Extension of Simplex Method

Let us define the quadratic programming problem

as:

$$\min. \left\{ f(x) \mid Ax \geq b, x \geq 0 \right\} ; \text{ where } f(x) = p^T x + \frac{1}{2} x^T C x ;$$

$C : nxn$ symmetric, non-negative definite, $A : mxn$; $p, x : nx1$, $b : mx1$. A simple example (written out in full) is given by:

$$\min. \left\{ (-44, -42) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \frac{1}{2} (x_1, x_2) \begin{pmatrix} 16 & -12 \\ -12 & 34 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \mid -2x_1 - x_2 \geq -10, \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\} .$$

The gradient of the quadratic objective function is given in the general case by

$$g(x) = \nabla f(x) = p + Cx$$

and in our example as

$$g(x) = \begin{pmatrix} -44 \\ -42 \end{pmatrix} + \begin{pmatrix} 16 & -12 \\ -12 & 34 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} ,$$

Our approach is to begin with an initial feasible solution (e.g. $x_1 = x_2 = 0$) and then increase some nonbasic variable x_ℓ if $\partial f / \partial x_\ell < 0$. In our example we choose to increase x_1 since $\partial f / \partial x_1 = -44 < 0$. An increase in the chosen nonbasic variable is useful until either $\partial f / \partial x_1$ or one of the basic variables becomes zero. To fix our ideas we illustrate the example by the following tableau:

x_1	x_2	b	
-8	6	-22	u_1
2	1	10	x_3
8	-6	-22	(x_1)
-6	17	-21	(x_2)
-22	-21	0	(1) .

From the above tableau we observe that the objective function can also be written as

$$f(x) = (-22, -21) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + (x_1, x_2) \begin{pmatrix} 8x_1 - 6x_2 - 22 \\ -6x_1 + 17x_2 - 21 \end{pmatrix} .$$

The initial feasible point at $x_1 = x_2 = 0$, $u_1 = -22$, $x_3 = 10$ gives an objective function value of $f(x) = 0$ (lower right hand corner of tableau). The initial basis consists of u_1 and x_3 . The constraints at the starting point are given in the upper part of the tableau, i.e.

$$\begin{aligned} u_1 &= -22 + 8x_1 - 6x_2, \text{ and} \\ x_3 &= 10 - 2x_1 - x_2 . \end{aligned}$$

We note that an increase in the nonbasic variable x_1 is useful until either u_1 or x_3 becomes zero. Since we keep x_2 fixed at zero, the first equation holds for a smaller x_1 . We now interchange variables x_1 and u_1 ; i.e. the basic variables become $x_1 = 2.75$, $x_3 = 4.5$ and the nonbasic variables are $x_2 = u_1 = 0$. This interchange of variables is accomplished in two steps; the first step is equivalent to the interchange of variables in the Simplex method; the second step is equivalent to the exchange of variables outside the brackets and does not present any difficulties. The new variable u_1 can take positive, zero, or negative values; hence, we call it a free variable.

We become interested in such a free variable u_1 when $\partial f / \partial u_1 \neq 0$. We introduce a new free variable $u_j = \partial f / \partial u_1$ and then proceed to make this new free variable nonbasic by interchanging the free variables u_1 and u_j . Once the

hitherto nonbasic free variable u_1 has been made basic again, it can be ignored, because it does not matter if it becomes negative through a change in the value of another variable, nor are we concerned with its value in the final solution.

We have not yet proved that our procedure terminates in a finite number of steps. The procedure is finite if a rule is followed concerning the choice of nonbasic variables to be made basic. This rule demands that if there exist any nonbasic free variables u_1 , such that $\partial f / \partial u_1 \neq 0$, then one of them must be made basic in a manner which reduces $f(x)$. A new nonbasic free variable will thus be introduced:

$$\partial f / \partial u_1 = u_j, \text{ say.}$$

An expression for $f(x)$ in which the linear terms contain no free variable is said to be in "standard form". To begin with, there exist no free variables at all. The following types of transformations can arise:

(i) introduction of a new nonbasic free variable, in exchange for a sign-restricted one (first cycle in the following example); it can be proved that the new variable will not appear in the linear term.

(ii) exchange of two sign-restricted variables (second cycle in the example); a linear term in a free variable can appear, but then it will lead, by our rule, to

(iii) the introduction of a new free variable, not in a linear term, and to the disappearance of a free

variable which had been in a linear term (third cycle in the example).

As long as there are linear terms in free variables, steps of type (iii) will be made. Hence, after a finite number of steps, we reach again a standard form. But there exist only a finite number and therefore after a finite number of steps, we reach a minimum.

This extension of the Simplex method to the quadratic programming problem is credited to Beale (1955); it is simple in principle and easy to execute by hand computation for the smaller problems. This review and the following example is based on the work by Vajda (1962).

§ 5.31.01 Example

Let us now consider the example first introduced in § 5.31 in greater detail; i.e.

$$\min. \left\{ (-44, -42) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \frac{1}{2} (x_1, x_2) \begin{pmatrix} 16 & -12 \\ -12 & 34 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \mid 2x_1 + x_2 + x_3 = 10, x \geq 0 \right\} .$$

The gradient of $f(x)$ is given by

$$g(x) = \nabla f(x) = \begin{pmatrix} -44 \\ -42 \end{pmatrix} + \begin{pmatrix} 16 & -12 \\ -12 & 34 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} .$$

The solution of this particular problem is obtained in three cycles as follows:

(i) introduction of a new non-basic variable u_1 in exchange for a sign-restricted x_1 ,

(ii) exchange of two sign-restricted variables x_2 and x_3 ,

(iii) introduction of a new free variable u_2 and the disappearance of a free variable u_1 .

The choice of variable which is to be introduced into the basis is determined by the composition of the linear term. We choose as our initial basis $x_3 = 10$; i.e. $x_1 = x_2 = 0$.

(1) An examination of the linear term with components $p_1 = -44$, $p_2 = -42$ reveals that both are negative. We choose - the choice is arbitrary - to increase the variable x_1 until either $g_1 = \partial f / \partial x_1$ becomes zero or the constraint $2x_1 + x_2 + x_3 = 10$ is violated, whichever occurs first. Here the first condition holds for a smaller x_1 . We introduce a new free variable into the basis by adding an additional constraint to our problem; i.e.

$$u_1 = \partial f / \partial x_1 = -22 + 8x_1 - 6x_2 ,$$

and increase x_1 until $u_1 = 0$.

The interchange problem (i.e. introduction of a sign-restricted variable into the basis in exchange for a free variable) denoted by

$$u_1 = -22 + 8x_1 - 6x_2 ,$$

$$x_3 = 10 - 2x_1 - x_2 ,$$

$$\begin{aligned} f(x_1, x_2) &= (-22 + 8x_1 - 6x_2) x_1 \\ &\quad + (-21 - 6x_1 + 17x_2) x_2 \\ &\quad + 0 - 22x_1 - 21x_2 \end{aligned}$$

is conveniently represented by the tableau

x_1	x_2	b	
-8	6	-22	u_1
2	1	10	x_3
8	-6	-22	(x_1)
-6	17	-21	(x_2)
-22	-21	0	(1)

The variables to be interchanged are x_1 and u_1 . The above tableau illustrates this choice by a negative coefficient of x_1 in the last row (the linear term) and the smallest positive ratio of the quantities in the b and x_1 columns of each row, i.e. $\min. (-22/8, 10/2)$. We observe that x_2 is held fixed at zero. Notice that in the same way as we would ignore a positive coefficient of x_1 in the constraint $x_3 = 10 - 2x_1 - x_2$, we could ignore a negative coefficient of x_1 in the expression $\partial f / \partial x_1$. Either case would indicate an unbounded value for $f(x)$.

The interchange of variables x_1 and u_1 is accomplished in two steps, first by interchanging inside the brackets using the Simplex method, second by interchanging outside the brackets. The interchange outside the brackets is accomplished by replacing the u_1^2 term by $1/\partial u_1 / \partial x_1$ and replacing all other terms in the bottom part of the tableau by zero.

Thus

x_1	x_2	b		u_1	x_2	b		u_1	x_2	b	
-8	6	-22	u_1	-.125	-.75	2.75	x_1				x_1
2	1	10	x_3	.25	2.5	4.5	x_3				x_3
8	-6	-22	$(x_1) \rightarrow$	1	0	0	$(x_1) \rightarrow$.125	0	0	(u_1)
-6	17	-21	(x_2)	-.75	12.5	-37.5	(x_2)	0	12.5	-37.5	(x_2)
-22	-21	0	(1)	0	-37.5	-60.5	(1)	0	-37.5	-60.5	(1)

The objective function can now be written as

$$f(u_1, x_2) = -60.5 + (0, -75) \begin{pmatrix} u_1 \\ x_2 \end{pmatrix} + \frac{1}{2} (u_1, x_2) \begin{pmatrix} .250 \\ 0.25 \end{pmatrix} \begin{pmatrix} u_1 \\ x_2 \end{pmatrix}.$$

We note that there is no linear term involving u_1 in the objective function.

(ii) An analysis of the linear term (the last row) now reveals it is worthwhile to increase x_2 , up to the smaller of $4.5/2.5$ (when x_3 becomes zero) and $37.5/12.5$, when $\partial f / \partial x_2 = -37.5 + 12.5 x_2$ becomes zero. The smaller of these ratios is the first, so that we interchange the two sign-restricted variables x_2 and x_3 . Thus

u_1	x_2	b		u_1	x_3	b		u_1	x_3	b	
-.125	-.75	2.75	x_1	-.05	.30	4.1	x_1				x_1
.25	2.5	4.5	x_3	.10	.40	1.8	x_2				x_2
.125	0	0	(u_1)	.125	0	0	(u_1)	0.25	0.5	1.5	(u_1)
0	12.5	-37.5	(x_2)	-1.25	-5	-15	(x_2)	0.5	2	6	(x_3)
0	-37.5	-60.5	(1)	3.75	15	-128	(1)	1.5	6	-155	(1) .

The above interchange is accomplished in a manner similar to (i) and will not be reviewed in detail. The objective function is now written as

$$f(u_1, x_3) = -155 + (3, 12) \begin{pmatrix} u_1 \\ x_3 \end{pmatrix} + \frac{1}{2} (u_1, x_3) \begin{pmatrix} .50 & 1 \\ 1 & 4 \end{pmatrix} \begin{pmatrix} u_1 \\ x_3 \end{pmatrix} .$$

We observe that we no longer have a "standard form" since a free variable is present in the linear term.

(iii) It is now necessary to return to a "standard form", i.e. we decrease u_1 - an increase is undesirable - until either

$$4.1 + 0.05u_1 = 0 , \text{ or}$$

$$1.5 + 0.25u_1 = 0 .$$

This happens the first time when $u_1 = -6$, and then the new variable

$$u_2 = 1.5 + 0.25u_1 + 0.5x_3$$

equals zero. We exchange the free variables u_1 and u_2 . Thus

u_1	x_3	b		u_2	x_3	b		u_2	x_3	b	
-.05	.30	4.1	x_1	-.2	.4	3.8	x_1				x_1
.10	1	1.8	x_2	.4	.2	2.4	x_2				x_2
-.25	-.5	1.5	$u_2 \rightarrow$	-1	2	-6	u_1				u_1
0.25	.5	1.5	(u_1)	1	0	0	(u_1)	4	0	0	(u_2)
0.5	2	6	(x_3)	2	1	3	(x_3)	0	1	3	(x_3)
1.5	6	-155	(1)	6	3	-164	(1)	0	3	-164	(1) .

Now an increase of x_3 or a change of u_2 cannot reduce the objective function

$$f(u_2, x_3) = -164 + (0, 6) \begin{pmatrix} u_2 \\ x_3 \end{pmatrix} + \frac{1}{2} (u_2, x_3) \begin{pmatrix} 8 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} u_2 \\ x_3 \end{pmatrix} ,$$

and we have reached the minimum with coordinates $x_1 = 3.8$,
 $x_2 = 2.4$ and function value $f = -164$.

That we are indeed at the optimum point is quite obvious since the coefficient of x_3 in the linear term is positive; the coefficient of u_2 in the linear term is zero; and both terms in the quadratic portion of $f(x)$ appear in the form of squares with positive coefficients. Clearly, we cannot increase x_3 or vary u_2 without increasing the value of the objective function.

§ 5.31.02 The Basic Computing Algorithm

We define

$$B \iff \left[\begin{array}{cc|c} \frac{1}{2}P^T : 1 \times n & & f(x) \\ \hline & & \\ \frac{1}{2}C : n \times n & \frac{1}{2}p : & \\ & n \times 1 & \\ \hline N^T : m \times n & b : m \times 1 & \end{array} \right] \quad D : (n+m+1) \times 1$$

$$\begin{array}{c} 0 \\ 1 \\ \dots \\ n \\ n+1 \\ \dots \\ n+m \end{array}$$

$$X \iff x(i) \iff x(j)$$

$$i=1, \dots, n \quad j=n+m+1$$

i.e. the introduction of a free variable in exchange for a sign-restricted one

$$Z \iff x(i) \iff x(j)$$

$$i=n+1, \dots, n+m \quad j=n+m+1$$

i.e. introduction of a new free variable in exchange for and

removal of a free variable.

$$f(x) = p^T x + \frac{1}{2} x^T C x ; g(x) = p + C x ; x : nx1.$$

comment

0;	$k \leftarrow 0$	
1;	$k \leftarrow i+1$	
	$i : n+1$, $\Rightarrow 2$ last term?
	$D(i) : n+1$, ≤ 1 is this a free variable?
	$B(1,i) : 0$, $\Rightarrow 1$ is coefficient zero?
	$B(n+m+2, \ell) \leftarrow -B(n+i+1, \ell)$	$\ell=1,2,\dots,n$
	$B(n+m+2, n+1) \leftarrow B(n+i+1, n+1)$	introduce free variable
	$Z \Longleftrightarrow \dots$	
	$\rightarrow 1$	
2;	$il \leftarrow \min_i B(1,i)$	$i=1,\dots,n$
	$B(1,il) : 0$, \geq END is sign + ?
	$\lambda_{il}^a \leftarrow \frac{-B(il+1, n+1)}{B(il+1, il)}$	
	$\lambda_k^b \leftarrow \min \left\{ \frac{B(n+i, n+1)}{B(n+i, il)} \right\}$	
	$\lambda \leftarrow \min (\lambda_j^b, \lambda_{il}^b)$	
3;	$X \Longleftrightarrow$	
	$\rightarrow 0$	
4;	$Y \Longleftrightarrow$	
	$\rightarrow 0$	

$$Y \Longleftrightarrow x(i) \Longleftrightarrow x(j) ; i=n+1,\dots,n+m, j=n+m+1$$

i.e. the exchange of two-sign restricted variables.

§ 5.32 Method of Feasible Directions

Let us continue our study of the quadratic programming problem by applying the method of feasible directions. We define our problem as follows:

maximize $\left\{ f(x) \mid Ax \geq b, x \geq 0 \right\}$, where $f(x) = p^T x - \frac{1}{2} x^T C x$;
 $C : n \times n$ symmetric, non-negative definite, $A : m \times n$; x , p :
 $n \times 1$, $b : m \times 1$.

The method of feasible directions is a method of steep descent and consists of determining a sequence of feasible solutions with ever-increasing values for the objective function. At each major cycle of our procedure we have to determine

(i) a feasible direction for which the objective function increases in value,

(ii) the length of the step to be taken in this direction.

The main contribution of the method of feasible directions to quadratic programming is the application of the Simplex method for solving the direction-finding problem.

If some of the linear constraints defined by $Ax \geq b$ are equalities instead of inequalities we can either eliminate some of the variables or we can always require $a_i s = 0$ instead of $a_i s \geq 0$ for the equality constraints $a_i x = b_i$ and $x \in R$. Elimination of variables reduces the size of the problem; but a simple form of A and $f(x)$ - if any exists - may be lost.

It is appropriate at this point to review the

notation applicable to this method. The function $e(x)$ is defined as $e(x) = Ax - b$ and clearly if $x^b \in R$ then $e(x^b) \geq 0$. A hyperplane is defined by $H_1 : e_1(x) = 0$ and the convention is used that the normal to the hyperplane points "into" the region R . The "best feasible" direction is that direction defined by the unit vector s and the gradient vector $g^b = g(x^b)$, i.e. the unit vector s for which

$$\begin{aligned} a_i s &\geq 0, \text{ if } e_i(x^b) = 0, \\ s^T s &= 1, \\ s^T g^b &\text{ is a maximum.} \end{aligned} \tag{1}$$

We assume here that $g^b \neq 0$; i.e. x^b is not the required solution.

Since we are interested only in those directions for which $f(x)$ increases in value, we must have $s^T g^b > 0$ and hence we can replace (1) by

$$\begin{aligned} a_i s &\geq 0, \text{ if } e_i(x^b) = 0, \\ s^T s &\leq 1, \\ s^T g^b &\text{ is a maximum.} \end{aligned} \tag{2}$$

Problem (2) has the same solution as (1) whenever $\max (s^T g^b) > 0$ in (1); and it has a solution with a value zero whenever $\max (s^T g^b) \leq 0$ holds in (1). The nonlinear constraint $s^T s \leq 1$ in (2) can be replaced by the weaker set of conditions that each component of s has to lie between $+1$ and -1 so that we obtain

$$\begin{aligned} a_i s &\geq 0, \text{ if } e_i(x^b) = 0, \\ -1 &\leq s_i \leq 1, \text{ all } i, \\ s^T g^b &\text{ is a maximum.} \end{aligned} \tag{3}$$

Although the solution of (3) in general differs from that in (2) we do, however, satisfy the condition $\max (s^T g^b) \geq 0$ in both cases. If $\max (s^T g^b) = 0$ holds in either (2) or (3) we shall not obtain a direct increase in $f(x)$ if we go from x^b in the direction s and for a concave $f(x)$ it follows that we have arrived at an absolute minimum. The direction-finding problem will now be deferred until we have considered a few more concepts.

Assuming we have available a feasible direction s^b which satisfies $(g^b)^T s^b > 0$, we want to make a step in that direction, so large that

(i) none of the constraints will be violated by the new trial solution $x = x^b + \lambda s^b$;

(ii) $f(x^b + \lambda s^b)$ will be maximized as a function of λ under the condition (i) above. There is a restriction in the choice of λ : since

$$e_1(x) = e_1(x^b + \lambda s^b) = e_1(x^b) + \lambda a_1 s^b \geq 0$$

must hold, we must choose $\lambda \leq \lambda^b$ where

$$\lambda^b = \min_i \left(\frac{e_1(x^b)}{|a_1 s^b|} \right), \text{ for those } i$$

for which $a_1 s^b < 0$. Hence we may reformulate the problem as

$\max_{0 \leq \lambda \leq \lambda^b} (f(x^b + \lambda s^b))$, for fixed x^b and s^b .

§ 5.32.01 Explicit Formulation of Method of Feasible Directions

The quadratic programming problem is defined as:

$$\text{maximize } \left\{ f(x) \mid Ax \geq b, x \geq 0 \right\}, \text{ where}$$

$$f(x) = p^T x - \frac{1}{2} x^T C x ;$$

$C : nxn$ symmetric non-negative definite,

$A : mxn$; $p, x : nx1$, $b : mx1$.

Using the relation

$$g(x) = p - Cx, \text{ we have}$$

$$\begin{aligned} \frac{\partial f}{\partial \lambda}(x+\lambda s) &= g(x+\lambda s)^T s \\ &= g^T s - \lambda s^T C s, \text{ so that} \end{aligned}$$

$$\frac{\partial f}{\partial \lambda}(x+\lambda s) = 0 \quad \text{if } \lambda = \lambda^a, \text{ defined by}$$

$$\lambda^a = \frac{g^T s}{s^T C s}.$$

(We have $s^T C s \geq 0$. If $s^T C s = 0$ then we put $\lambda^a = +\infty$, since $f(x)$ is then linear along the line $x = x^b + \lambda s$.)

It follows that

$$\begin{aligned} \lambda &= \min (\lambda^a, \lambda^b) \text{ where} \\ \lambda^b &= \min_i \left(\frac{e_i(x^b)}{|a_i^T s|} \right), \text{ for those } i \text{ for which } a_i^T s < 0. \end{aligned}$$

The iteration formulae are :

$$\begin{aligned} x^{k+1} &= x^k + \lambda_k s^k, \\ g^{k+1} &= g^k - \lambda_k C s^k, \quad k = 1, 2, \dots, n, \dots, \text{ and} \end{aligned}$$

$$\begin{aligned} \Delta f(x^k) &= \lambda_k (g^k)^T s^k - \frac{1}{2} \lambda_k^2 (s^k)^T C s^k, \\ &= \frac{1}{2} \lambda_k \left\{ (g^k)^T + (g^{k+1})^T \right\} s^k, \\ &= \frac{1}{2} \lambda_k (g^k)^T s^k, \text{ if } \lambda_k = \lambda^a. \end{aligned}$$

The procedure in general is not finite in the quadratic case. It can easily be made finite by using the theory for solving linear equations as developed by Hestenes and Stiefel (1952). Clearly, the problem of finding the unrestricted maximum of the strictly concave quadratic function $f(x) = p^T x - \frac{1}{2} x^T C x$ is equivalent to solving the linear system $Cx = p$.

Let x^0 and an initial direction s^0 define a second approximation x^1 of the solution α by means of

$$x^1 = x^0 + \lambda_0 s^0 \text{ with}$$

$$\lambda_0 = \frac{(g^0)^T s^0}{(s^0)^T C s^0} \quad \text{and } g^0 = p - Cx^0.$$

Next we choose a new direction s^1 which is conjugate to s^0 , in the sense that

$$(s^1)^T C s^0 = 0.$$

Having obtained the approximation x^k of α and put $g^k = p - Cx^k$, we choose a new direction s^k which is conjugate to s^0, s^1, \dots, s^{k-1} , so that

$$(s^k)^T C s^j = 0 \quad (j = 0, 1, \dots, k-1), \quad (1)$$

and we define $x^{k+1} = x^k + \lambda_k s^k$ with $\lambda_k = \frac{(g^k)^T s^k}{(s^k)^T C s^k}$.

The conditions (1) do not determine s^k uniquely. One method of choosing the vector s^k is to require $(g^k)^T s^k$ maximum and $(s^k)^T s^k \leq 1$, (or $-1 \leq s^k \leq 1$). This method will lead to the solution α in a finite number $r \leq n$ of steps.

If we have to maximize a concave quadratic function $f(x) = p^T x - \frac{1}{2} x^T C x$, subject to a set of linear inequalities $a_i x \geq b_i$ ($i=1,2,\dots,m$), the procedure of Hestenes and Stiefel cannot be applied without modification since the point $\alpha \notin R$ (generally). The modified procedure is then as follows: we assume that we are on the correct set of hyperplanes (the set in which the maximum lies); and proceed to conjugate successive directions s^0, s^1, \dots, s^{k-1} , either until we arrive at a step-length defined by λ_i^b or until we arrive at a maximum. In the former case our assumption is false; we drop the additional requirements since we are not working on the correct set of hyperplanes and add the hyperplane determined by λ_i^b to our set of hyperplanes on which we assume our maximum lies and add the constraint $a_i s \geq 0$ since $e_i(x) = 0$. We now start conjugation of successive directions s^0, s^1, \dots, s^{k-1} , as before.

§ 5.32.02 Determination of Optimum Feasible Directions

The direction-finding problem is defined as

$$B s \geq 0 ,$$

$$s^T s \leq 1 , \quad (1)$$

$$g^T s \text{ is a maximum,}$$

$B : m \times n$; $s, g : n \times 1$. The matrix B has n columns and m rows corresponding to constraints in which the trial solution lies.

If $\max (g^T s) > 0$ we may normalize and restate

our direction-finding problem as

$$\begin{aligned} B s &\geq 0, \\ g^T s &= 1, \\ s^T s &\text{ is a minimum.} \end{aligned} \quad (2)$$

This problem can now be solved as a standard quadratic programming problem but the following special technique is available.

According to the solution criterion of Kuhn and Tucker (1951) the gradient at the optimum point must be a non-negative linear combination of the outward-pointing normals in that point, i.e. an optimal vector s of (1) must also satisfy the relations

$$\begin{aligned} g &= -B^T u + 2\beta s, \text{ where } u \text{ are Lagrange Multipliers} \\ u^T B s &= 0 \\ u &\geq 0; \end{aligned} \quad (3)$$

$u : mx1$; β : a scalar. The above result is also immediately available from the Farkas Theorem (§ 4.32.01) if we replace A by B . Hence a normal may not be taken into account if the optimum point does not lie in the corresponding constraints, so that $u_i = 0$ holds if $b_i s < 0$, i.e. $u_i^T b_i s = 0$, where b_i is the i -th row of B .

Multiplying $g = -B^T u + 2\beta s$ by $-B$ and defining $2\beta B s = v$, we obtain

$$\begin{aligned} BB^T u - v &= -Bg, \\ u^T v &= 0, \\ u &\geq 0, v \geq 0; \quad v : mx1. \end{aligned} \quad (4)$$

Having found such vectors u and v , we obtain $2\beta s$ by the

relation $2\beta s = g + B^T u$, and s by normalizing although this normalization is not necessary in the direction-finding problem. The event $\beta = 0$ (hence $\beta s = 0$) leads to $g^T s = -u^T B s = 0$, so that we have then arrived at a stationary point.

Consider (4) and start with $u = 0$, $v = Bg$. If $v \geq 0$ holds, we have the required solution; otherwise, we take a negative v_j and carry out a transformation of the Dual Simplex algorithm by taking the diagonal element in that row as the pivot; we thus satisfy the relation $u^T v = 0$ automatically throughout the calculation. Cycling can be prevented in this procedure by choosing the negative right-hand member in a special simple way. If we are dealing with equalities instead of inequalities the corresponding y_i 's and hence also the v_i 's have to be zero; the corresponding u_i 's are no longer sign-restricted in this case. We can thus start by eliminating the v_i 's involved from the basis (replace them by the corresponding u_i 's).

Again, the starting point of the direction-finding problem can be described by the tableau:

$u_1 \dots u_m$		
- BB^T	$b_1 g$	v_1
	$b_m g$	v_m
0 0		,

where b_j is the j -th row of B .

§ 5.33 Gradient Projection Method

Let us define our programming problem as:
 maximize $\left\{ f(x) \mid N^T x \leq b, x \geq 0 \right\}$, where $f(x) = p^T x + \frac{1}{2} x^T C x$,
 and the set of m vectors defined by $N_m \equiv [n_1, n_2, \dots, n_m]$
 is normalized so that $n_i^T n_i = 1$, $i=1, 2, \dots, m$; $C : nxn$
 symmetric, non-negative definite, $N : nxm$; $p, x, n_1, n_2, \dots,$
 $n_m : nx1, b : mx1$.

The gradient projection method starts with a feasible point $x^0 \in R$, where R is defined by the constraints $N_m^T x \leq b$, and proceeds in a direction defined by the gradient of the objective function projected into the space which is the intersection of all hyperplanes

$$Q : H_i : n_i^T x - b_i = 0, i = 1, 2, \dots, q,$$

which are currently satisfied as equalities; if this projection is not positive, then either the optimum has been reached (and criteria are given to determine whether this has happened), or one of the constraints is ignored and we proceed along the component of the gradient of the remaining intersection space. A positive projection of the gradient will increase the value of the objective function and we proceed along it.

The contents of this chapter are divided into four sections. In section one we consider the calculational problems involved in the gradient projection method. During an optimization calculation it is necessary to form a square matrix $N^T N$ and the inverse matrix $(N^T N)^{-1}$ from N

a matrix whose columns are a set of independent vectors. A new inverse obtained by adding or deleting one column of N is usually required at each step. The computing algorithm is used freely to describe the recursive relations which give the new inverse and corresponding projection matrix. The step-size problem is also reviewed in this first section.

In sections 2 and 3 we define the basic computing algorithm for the gradient projection and solve an elementary example. In the last section we consider some difficulties which may arise in carrying out the calculations of the preceding sections; these difficulties include linear dependence of constraints, rounding error, solution improvement, and the application of the gradient projection method to the linear programming problem. In preparing this chapter, we have made use of earlier work by Rosen (1960).

§ 5.33.01 Preview of Calculations

In the course of an optimization calculation using the Gradient Projection method it is necessary to obtain the projection of the gradient vector on various intersections Q . Each new intersection will be determined by a set of hyperplanes which differs from the previous set by one. For example, one of the hyperplanes may be dropped from the set; a hyperplane may be added to the set; a hyperplane may be dropped and another one added; and finally the set may remain unchanged. We may note that

there appears to be a formal similarity between an interchange of hyperplanes in the gradient projection method and an interchange of variables in the Simplex method. For the purposes of the following discussion on the calculational problems involved in the Gradient Projection method we shall assume that the inverse matrix $(N_q^T N_q)^{-1}$ is known.

The recursive relations used which permit a hyperplane to be dropped from $(N_q^T N_q)^{-1}$ with approximately q^2 multiplications, and a hyperplane to be added with approximately $3qn$ multiplications are based on the formula for the inverse of a matrix in terms of the inverses of certain of its partitions. In particular, suppose a square matrix A is partitioned as

$$A = \begin{bmatrix} A_1 & : & A_2 \\ \dots & : & \dots \\ A_3 & : & A_4 \end{bmatrix} \quad (1)$$

where A_1 and A_2 are square matrixes and A_2 , A_3 are rectangular matrices with the appropriate number of rows and columns.

Then if both A_1 and

$$A_0 = A_4 - A_3 A_1^{-1} A_2 \quad (2)$$

are nonsingular, the inverse of A exists and is given by

$$A^{-1} = B = \begin{bmatrix} B_1 & : & B_2 \\ \dots & : & \dots \\ B_3 & : & B_4 \end{bmatrix} \quad (3)$$

where

$$\begin{aligned} B_4 &= A_0^{-1}, \\ B_3 &= -B_4 A_3 A_1^{-1}, \\ B_2 &= -A_1^{-1} A_2 B_4 \\ B_1 &= A_1^{-1} - B_2 A_3 A_1^{-1}. \end{aligned} \quad (4)$$

For computational efficiency we note that the desired quantities should be obtained in the specific order, A_1^{-1} , $A_3 A_1^{-1}$, A_0 , B_4 , B_3 , B_2 , and B_1 . An expression for A_1^{-1} is readily available and given by

$$A_1^{-1} = B_1 - B_2 B_4^{-1} B_3 . \quad (5)$$

The relation (5) will be applied to the following problem. Given a set of q linearly independent hyperplanes and the corresponding inverse matrix $(N_q^T N_q)^{-1}$, to drop one of the hyperplanes from the set and to obtain the corresponding inverse matrix for the $(q-1)$ hyperplanes. In particular, we assume $(N_q^T N_q)^{-1}$ is known and that H_q is to be dropped and $(N_{q-1}^T N_{q-1})^{-1}$ obtained, where

$$N_{q-1} \equiv [n_1, n_2, \dots, n_{q-1}] . \quad (6)$$

$$\begin{aligned} \text{Let} \quad A &= N_q^T N_q & ; \quad A : qxq , \\ A_1 &= N_{q-1}^T N_{q-1} & ; \quad A_1 : (q-1)x(q-1) . \end{aligned} \quad (7)$$

We assume that $B = A^{-1} = (N_q^T N_q)^{-1}$ is known and therefore that the partitions B_1 , B_2 , B_3 , and B_4 of B are also known. In particular, B_1 is a symmetric $(q-1)x(q-1)$ matrix, B_2 a $(q-1)$ - dimensional column vector, $B_3 = B_2^T$, and B_4 a scalar. The required inverse $(N_{q-1}^T N_{q-1})^{-1} = A_1^{-1}$ is then given by (5). If H_ℓ is to be dropped, the relation (5) applies if the ℓ -th and q -th row and column of $(N_q^T N_q)^{-1}$ are interchanged before it is applied.

The procedure for adding a hyperplane to the inverse is now described. For this purpose it is assumed

that $(N_{q-1}^T N_{q-1})^{-1}$ is known and it is desired to add H_q to the inverse matrix. It follows from (7) that

$$\begin{aligned} A_2 &= N_{q-1}^T n_q = A_3^T, \\ A_4 &= n_q^T n_q = 1. \end{aligned} \quad (8)$$

A_2 is a $(q-1)$ -dimensional column vector, and $A_4=1$ follows from the normalization of the hyperplanes. From (2), (7), and (8) it follows that

$$\begin{aligned} A_0 &= n_q^T n_q - N_{q-1} (N_{q-1}^T N_{q-1})^{-1} N_{q-1}^T n_q \\ &= n_q^T (I - N_{q-1} (N_{q-1}^T N_{q-1})^{-1} N_{q-1}^T) n_q \\ &= n_q^T P_{q-1} n_q. \end{aligned} \quad (9)$$

It can be shown that A_0 is also defined by

$$A_0 = (P_{q-1} n_q)^T (P_{q-1} n_q) = |P_{q-1} n_q|^2. \quad (10)$$

A necessary and sufficient condition that $A_0 > 0$, is that H_q is linearly independent of the intersection $H_i, i=1,2,\dots, q-1$. Assuming this to be the case, (4) gives

$$\begin{aligned} B_1 &= (N_{q-1}^T N_{q-1})^{-1} + A_0^{-1} r_{q-1} r_{q-1}^T, \\ B_2 &= A_0^{-1} r_{q-1} = B_3^T, \end{aligned} \quad (11)$$

$$\begin{aligned} B_4 &= A_0^{-1}, \text{ where} \\ r_{q-1} &= (N_{q-1}^T N_{q-1})^{-1} (N_{q-1}^T n_q). \end{aligned} \quad (12)$$

We also note that

$$P_{q-1} n_q = n_q - N_{q-1} r_{q-1}. \quad (13)$$

A very useful recursive relation is available between P_q , P_{q-1} , and n_q . Again let $(N_q^T N_q)^{-1} = B$ as in (3)

and the B_i , $i=1,2,3,4$, as given in (11). Comparing (10) and (6), the matrix N_q can be written $N_q = [N_{q-1}, n_q]$ and similarly for its transpose. From its definition, \bar{P}_q is therefore given by

$$\bar{P}_q = [N_{q-1}, n_q] B \begin{bmatrix} N_{q-1}^T \\ n_q^T \end{bmatrix}. \quad (14)$$

After carrying out the matrix multiplications and substituting for B from (11), the following is obtained,

$$\bar{P}_q = \bar{P}_{q-1} + A_o^{-1} (P_{q-1} n_q) (P_{q-1} n_q)^T. \quad (15)$$

Let u_q denote the unit vector parallel to $P_{q-1} n_q$,

$$u_q = P_{q-1} n_q / |P_{q-1} n_q|. \quad (16)$$

Subtracting both sides of (15) from I : $n \times n$ unit matrix gives

$$P_q = P_{q-1} - u_q u_q^T \quad (17)$$

The recursion relations which have just been described also make it possible to build up the matrices $(N_\ell^T N_\ell)^{-1}$ and P_ℓ for a set of ℓ linearly independent vectors u_i , $i = 1, 2, \dots, \ell$, with a minimum of computation.

The problems of adding and subtracting a hyperplane from $(N_q^T N_q)^{-1}$ and the construction of the associated projection matrix will now be described by computing algorithms.

Computing Algorithm - Adding a Hyperplane

The following algorithm performs the operations defined as:

$$\left(\begin{array}{l} (N_q^T N_q)^{-1} \longleftarrow (N_{q-1}^T N_{q-1})^{-1} \\ P_q \longleftarrow P_{q-1} \end{array} \right).$$

For purposes of ease of description we also define

$$(B \iff \begin{bmatrix} B_1 & \vdots & B_2 \\ \dots & \dots & \dots \\ B_3 & \vdots & B_4 \end{bmatrix} \iff (N_q^T N_q)^{-1}) .$$

$t \leftarrow N_{q-1}^T n_q$	$t: (q-1) \times 1$
$r_{q-1} \leftarrow (N_{q-1}^T N_{q-1})^{-1} t$	
$t \leftarrow n_q - N_{q-1} r_{q-1}$	$t: n \times 1$
$A_0 \leftarrow t^T t$	$A_0: \text{scalar},$ $A_0 > 0$
$B_4 \leftarrow 1/A_0$	
$B_2 \leftarrow -B_4 r_{q-1}$	$B_2: (q-1) \times 1$
$B_3 \leftarrow B_2^T$	
$B_1 \leftarrow (N_{q-1}^T N_{q-1})^{-1} - B_2 r_{q-1}^T$	
$u_q \leftarrow B_4 t$	$u_q: n \times 1$
$P_q \leftarrow P_{q-1} - u_q u_q^T$	$P_q: n \times n$
END	

Computing Algorithm - Subtracting a Hyperplane

The operations performed by this algorithm are defined as:

$$\left(\begin{array}{l} (N_{q-1}^T N_{q-1})^{-1} \leftarrow (N_q^T N_q)^{-1} \\ P_{q-1} \leftarrow P_q \end{array} \right)$$

$$\begin{aligned} (N_{q-1}^T N_{q-1})^{-1} &\leftarrow B_1 - \left(\frac{1}{B_{qq}}\right) B_2 B_2^T \\ A_2 &\leftarrow N_{q-1}^T n_q \quad . \quad A_2: (q-1) \times 1 \end{aligned}$$

$$\begin{aligned} r_{q-1} &\leftarrow (N_{q-1}^T N_{q-1})^{-1} A_2 \\ u_q &\leftarrow n_q - N_{q-1} r_{q-1} \\ P_{q-1} &\leftarrow P_q + u_q u_q^T \end{aligned}$$

Computing Algorithm - Building the Projection Matrix

The purpose of this algorithm is to construct from ℓ linearly independent normalized vectors

$$N_\ell \equiv [n_1, n_2, \dots, n_\ell]$$

the following:

$$\left(\begin{array}{l} B \iff (N_\ell^T N_\ell)^{-1}, \text{ and} \\ \bar{P}_\ell \iff I - N_\ell (N_\ell^T N_\ell)^{-1} N_\ell^T \end{array} \right)$$

comment

```

1 ← 1
B ← 1
N ← n1
P1 ← I - n1 n1T

1;   i ← i+1
      t ← NT ni           t: scalar for i=2
      r ← Bt
      t ← Nr
      t ← ni - t           t: nxl
      A0 ← tT t           if A0 = 0 then ni lin. dep.
      B4 ← 1/A0
      B1 ← B4 r rT
      B2 ← -B4 r
      B3 ← B2T
      u ← t / |t|
      P ← P - u uT
      ℓ : i , ≥ 1           all vectors included?

```


During an optimization calculation we must always solve the step-size problem. We can assume that we have available the gradient of $f(x)$ i.e., $g(x^k) = p + Cx^k$ and the projection matrix P_q . The necessary and sufficient conditions (theorem 2) that x^k be a constrained global maximum are $P_q g(x^k) = 0$, and

$$r = (N_q^T N_q)^{-1} N_q g(x^k) \leq 0 ; r : qx1 .$$

If we are not at a maximum we test r and drop that hyperplane defined by

$$\max_i \left\{ r_i \right\} > 0$$

from the projection matrix. In this way we obtained a revised projection matrix P . The manifold Q has a dimensionality of one (a line) at least. We define the unit vector

$$s = P_q g(x^k) / | P_q g(x^k) | .$$

It follows that $N_q^T s = 0$, so that $x = x^k + \lambda s$ is in Q for any λ . By assumption, x^k lies on only the q hyperplanes in Q , H_i , $i=1, 2, \dots, q$, and since it is a feasible point, $e_i(x^k) \geq \delta > 0$, $i=q+1, \dots, m$. Then for each of the remaining $m-q$ hyperplanes, H_i , $i=q+1, \dots, m$, there may exist a value $\lambda = \lambda_i$ such that $e_i(x) = 0$. $|\lambda_i|$ is the distance from x^k to the hyperplane H_i ; the distance from x^k to the hyperplane H_i is along a parallel to s . In particular, we shall define λ^b as the minimum quantity chosen from the set of λ_i which are positive,

$$\text{i.e. } \lambda^b = \min \left\{ \lambda_i = e_i(x^k) / s^T n_i \geq \delta > 0 \right\} ,$$

if $s^T n_i > 0$ and $i = q+1, \dots, m$. The distance λ^b represents the largest step that can be taken from x^k in the direction s without leaving R (the region defined by the constraints

$$N_m^T x \leq b).$$

A further restriction on the step-length is that we want to maximize $f(x^k + \lambda s)$ as a function of λ , i.e.

$$\begin{aligned} \frac{\partial f}{\partial \lambda}(x + \lambda s) &= (Pg(x^k + \lambda s))^T s \\ &= g^T Ps - \lambda s^T C Ps, \end{aligned}$$

and for a maximum step-size we have

$$\lambda^a = g^T s / s^T C s, \text{ since } Ps = s.$$

Hence, the step-size used is given by $\lambda = \min(\lambda^a, \lambda^b)$.

Computing Algorithm - Step-size Problem

We wish to find λ defined by:

$$\lambda = \min(\lambda^a, \lambda_j^b)$$

comments

<pre> t ← Pg(x^k) s ← t / t t ← Cs b ← s^Tt λ^a ← g^Tt / b i ← q λ^b ← L 1; i ← i+1 b ← n_i^Ts b : 0 b ← e_i(x^k) / -b b : 0 λ^b : b λ^b ← b j ← i m : i λ ← min(λ^a, λ_j^b) END </pre>	<pre> , ≥ 1 , → 1 , ≤ 1 , → 1 , ≥ 1 </pre> <p>we choose L very large</p>
---	--

$(N_q^T N_q)^{-1} \leftarrow (N_{q-1}^T N_{q-1})^{-1} \quad \text{add a hyperplane}$ $P_q \leftarrow P_{q-1}$ $\rightarrow 1$ END

§ 5.33.03 Example of Gradient Projection Method

The procedures and algorithms defined in the previous sections will now be applied to the following quadratic programming problem:

$$\max \left\{ f(x) = p^T x + \frac{1}{2} x^T C x \mid N_7^T x \leq b, x \geq 0 \right\} ;$$

where $f(x) = (3, 0, 4) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \frac{1}{2} (x_1, x_2, x_3) \begin{pmatrix} 4 & -2 & 1 \\ -2 & 3 & 0 \\ 1 & 0 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} ,$

and $N_7 \equiv \begin{bmatrix} 1 & 0 & 0 & -1/\sqrt{2} & -1 & 0 & 0 \\ 0 & 1 & 0 & -1/\sqrt{2} & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} .$

We note that

$$P = \begin{pmatrix} 3 \\ 0 \\ 4 \end{pmatrix} , \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} , \quad C = \begin{pmatrix} 4 & -2 & 1 \\ -2 & 3 & 0 \\ 1 & 0 & 5 \end{pmatrix} ;$$

$C : 3 \times 3$, symmetric non-negative definite. We shall start with the initial feasible point $(x^0)^T = (0.1, 0.1, 0.1)$. The initial point x^0 is feasible and does not lie on any hyperplanes since

$$e_i(x^0) = n_i^T x^0 - b_i > 0 , \quad i = 1, 2, \dots, 7 .$$

Clearly, since the initial point is feasible and not constrained to lie on any hyperplanes we have an initial gradient vector $g(x^0) = p + Cx^0$, an initial projection matrix $P_0 = I_3 : 3 \times 3$ identity matrix, and a unit direction vector

$s = Pg^0 / |Pg^0| = g^0 / |g^0|$. Evaluating the required functions at the initial value $(x^0)^T = (0.1, 0.1, 0.1)$ we obtain the following:

$$g^0 = g(x^0) = \begin{pmatrix} 3.3 \\ 0.1 \\ 4.6 \end{pmatrix} , P_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} , \text{ and } s = \begin{pmatrix} .58 \\ .02 \\ .81 \end{pmatrix} .$$

This particular problem will be solved in three cycles. Each cycle will start with a test for the optimum point and end with the definition of a new projection matrix.

(1) To start, the point x^0 is not the required optimum point since

$$P_0 g^0 = g^0 \neq 0 .$$

We now need to determine the minimum step-size defined by

$$\lambda = \min (\lambda^a, \lambda^b) .$$

We recall that

$\lambda^a = g^T s / s^T C s$, $\lambda^b = \min_i \left\{ \lambda_i = e_i(x^0) / s^T n_i \geq \delta > 0 \right\}$, where $e_i(x^0) = n_i^T x^0 - b_i$, and only those i are considered for which $s^T n_i > 0$ and $i = q + 1, \dots, m$. In our case we must test for every λ_i , i.e. $i = 1, 2, \dots, 7$. We obtain

$$\lambda^b = \min \left\{ \lambda_i = \lambda_7 = 1.11 \right\} .$$

Hence our step-size problem has the solution

$$\lambda = \min (5.69 , 1.11_7) ,$$

where the subscript 7 is used to denote the result that hyperplane H_7 will need to be added to our projection matrix.

The next approximate solution is given by

$$x^1 = x^0 + \lambda s = \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \end{pmatrix} + 1.11 \begin{pmatrix} .58 \\ .02 \\ .81 \end{pmatrix} = \begin{pmatrix} .74 \\ .12 \\ 1.00 \end{pmatrix} .$$

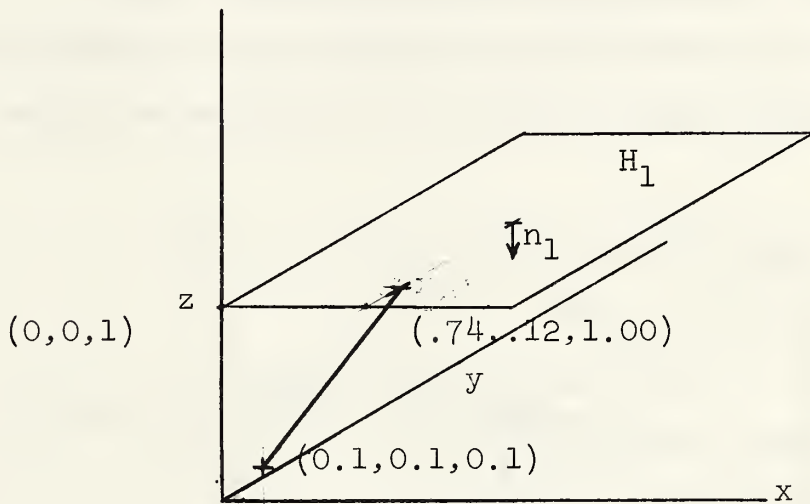
Since the constraint $e_7 = n_7^T x^0 - b_7$ determined the step-size we include H_7 in the projection matrix. For convenience we interchange constraints 1 and 7, i.e.

$$N_7 = \begin{bmatrix} 0 & 0 & 0 & -1/\sqrt{2} & -1 & 0 & 1 \\ 0 & 1 & 0 & -1/\sqrt{2} & 0 & -1 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} .$$

The intersection Q now consists of H_1 and the projection matrix P_1 is given by

$$P_1 = P_0 - n_1 n_1^T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} .$$

The following diagram depicts the constraint upon which we continue our search for the optimum point and the path followed in reaching the constraint.



(ii) We again follow the basic computing algorithm and test for the optimum point, i.e.

$$g^1 = g(x^1) = \begin{pmatrix} 4.56 \\ -1.12 \\ 9.74 \end{pmatrix} , \quad P_1 g^1 = \begin{pmatrix} 4.56 \\ -1.12 \\ 0 \end{pmatrix} , \quad r_\ell = -9.74 .$$

Clearly, we are not at the optimum point and furthermore we cannot drop our hyperplane from the projection matrix since $r_\ell < 0$.

The step-size problem is now solved and we obtain

$$\lambda = \min (6.63, .01_6) .$$

The next approximate solution is given as

$$x^2 = x^1 + \lambda s = \begin{pmatrix} .74 \\ .12 \\ 1.00 \end{pmatrix} + .01 \begin{pmatrix} .97 \\ -.24 \\ .00 \end{pmatrix} = \begin{pmatrix} .75 \\ .12 \\ 1.00 \end{pmatrix} .$$

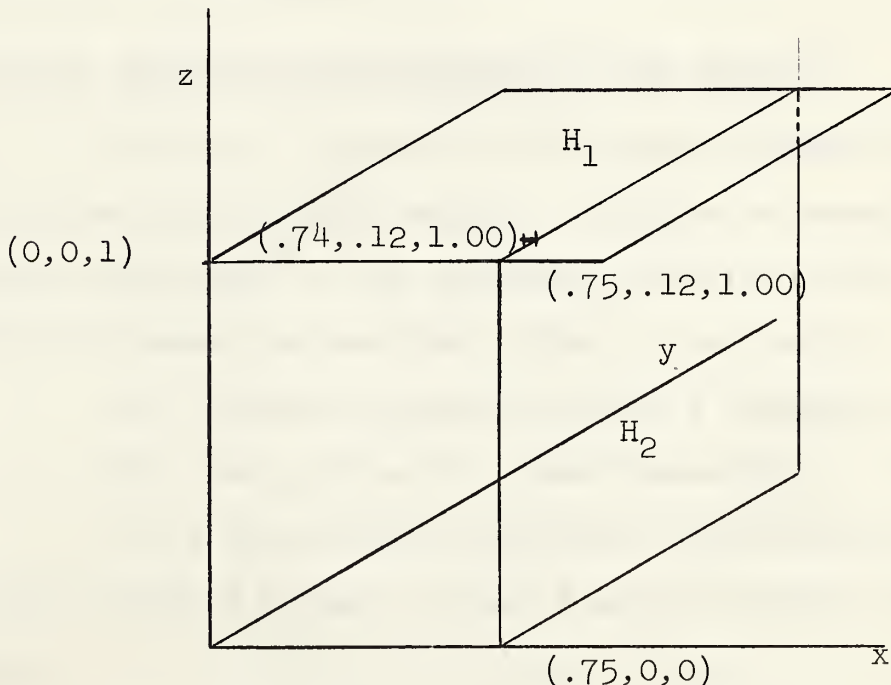
Again we interchange constraints 2 and 6 obtaining

$$N_7 = \begin{bmatrix} 0 & 0 & 0 & -1/\sqrt{2} & -1 & 0 & 1 \\ 0 & -1 & 0 & -1/\sqrt{2} & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} .$$

The intersection Q now consists of H_1 and H_2 and the projection matrix P_2 becomes

$$P_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} .$$

The following diagram now describes the constraint (now a straight line defined by the intersection of H_1 and H_2) upon which we continue our search for the optimum point.



(iii) From the basic computing algorithm we

obtain

$$g^2 = g(x^2) = \begin{pmatrix} 6.76 \\ -1.14 \\ 9.75 \end{pmatrix}, \quad p_2 g^2 = \begin{pmatrix} 0 \\ -1.14 \\ 0 \end{pmatrix}, \quad r_\ell = -9.75.$$

Again we continue our search by solving the step-size problem, obtaining

$$\lambda = \min (5.65, .126) .$$

Our next approximate solution becomes $x^3 = x^2 + \lambda s$

$$= \begin{pmatrix} .75 \\ .12 \\ 1.00 \end{pmatrix} + .12 \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} = \begin{pmatrix} .75 \\ 0 \\ 1.00 \end{pmatrix} .$$

We interchange constraints 3 and 6 and construct

$$p_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} .$$

Clearly, we are now at the optimum point with coordinates $x^T = (.75, 0, 1.00)$ since

$$p_3 g(x^3) = 0, \text{ and} \\ r_\ell = -1.50 < 0 .$$

§ 5.33.04 Further Considerations on the Method

In this - the last of the four sections mentioned in the introduction to this chapter - section we consider some important additions to the gradient projection method which greatly increase the practical value of the method. These are:

- (i) a starting procedure from a nonfeasible point,
- (ii) the problem of linear dependence on constraints,
- (iii) a correction procedure for improvement of the solution required because of build up of rounding errors, and finally

- (iv) an evaluation of the gradient projection method applied to the linear programming problem.

(i) We first consider the problem of a starting procedure for an arbitrary nonfeasible point x^0 which is defined by $e(x^0) \leq 0$, $e : q \times 1$, (we assume that at least one constraint is violated, i.e. $e_i(x^0) < 0$, for some i , $i=1, 2, \dots, q$) . For simplicity, we assume that the planes are ordered so that

$$e_i \quad \begin{cases} \leq 0 & (i = 1, 2, \dots, q) \\ > 0 & (i = q+1, \dots, m) \end{cases} .$$

and that the q hyperplanes H_i , $i=1, 2, \dots, q$, are linearly independent. We define the $q \times 1$ vector

$$e(x) \equiv \left\{ e_1(x), \dots, e_q(x) \right\} .$$

By definition, we have that

$$e(x) = N_q^T x - b_q ; \quad b_q : q \times 1 \quad . \quad (1)$$

$$\text{Let} \quad x^1 = x^0 - N_q (N_q^T N_q)^{-1} e(x^0) . \quad (2)$$

Hence from (1) we have

$$e(x^1) = e(x^0) - N_q^T N_q (N_q^T N_q)^{-1} e(x^0) = 0 \quad . \quad (3)$$

Thus x^1 , computed in (2), lies in all q of the selected hyperplanes. If in addition we have

$$e_i(x^1) \geq 0 \quad (i=q+1, \dots, m)$$

then x^1 is the desired starting point.

If at least one of the quantities $e_i(x_1)$, $i=q+1, \dots, m$, is negative, a minimum of these is chosen, say $i=q+1$. The projection $P_q n_{q+1}$ of n_{q+1} on the intersection Q of H_i , $i=1, \dots, q$, and the corresponding $r : q \times 1$ are then obtained from

$$r = (N_q^T N_q)^{-1} (N_q^T n_{q+1}) ,$$

$$P_q n_{q+1} = n_{q+1} - N_q r .$$

There are now three relevant possibilities.

1. $|P_q n_{q+1}| = 0$, $r_i < 0$, $i=1, \dots, q$. In this case there is no feasible solution, since it can be proved that no step is possible in a direction to increase $e_{q+1}(x)$ without violating at least one of the constraints in Q .

2. $|P_q n_{q+1}| = 0$, $r_i > 0$, for at least one i , say $i=1$. In this case drop n_1 and add n_{q+1} to N_q , giving $N_q^1 = [n_2, \dots, n_{q+1}]$. It can be shown that $N_q^1 N_q^1{}^T$ is not singular. Also, define $e_q^1(x^1) : q \times 1$ where

$$e_q^1(x^1) \equiv \left\{ 0, \dots, 0, e_{q+1}(x^1) \right\}.$$

A point x^2 is then obtained from

$$x^2 = x^1 - N_q^1 (N_q^1 N_q^1{}^T)^{-1} e_q^1(x^1).$$

It can be shown that

$$e_1(x^2) > 0,$$

$$e_i(x^2) = 0, \quad (i=2, \dots, q+1).$$

3. $|P_q n_{q+1}| > 0$. Add n_{q+1} to N_q , giving $N_{q+1} = [n_1, \dots, n_{q+1}]$ and obtain $(N_{q+1}^T N_{q+1})^{-1}$ by the recursion relation.

Also define $e_{q+1}(x) : q+1 \times 1$ where

$$e_{q+1}(x^1) \equiv \left\{ 0, \dots, 0, e_{q+1}(x^1) \right\}.$$

A point x^2 , satisfying $e_i(x^2) = 0$, $i=1, \dots, q+1$, is then given by

$$x^2 = x^1 - N_{q+1} (N_{q+1}^T N_{q+1})^{-1} e_{q+1}(x^1).$$

Repeated applications of the above procedure will either obtain a feasible point using (2) or (3), or show that no such point exists as in (1).

(ii) At this point we have assumed that all hyper-planes used were linearly independent. In general, this will not be the case either in the application of the gradient

projection method or in other applications where the projection matrix P or the inverse $(N^T N)^{-1}$ is needed. Given an arbitrary set of ℓ vectors n_1 , the recursion relations give a practical method of picking a largest linearly independent subset of these, i.e., a subset of ℓ' vectors which span the ℓ' subspace of the given set of ℓ vectors. If, e.g. in (11) we find that $P_{q-1} n_q = 0$, then we delete n_q from the set of ℓ vectors. It can be shown that this process if continued gives the desired subset of $\ell' \leq \ell$ vectors; the matrix $P_{\ell'}$ obtained in this way takes any vector into the intersection of the set of all the original ℓ hyperplanes, Rosen (1960).

(iii) A correction procedure may be needed in the gradient projection method due to the build up of rounding errors in the inverse matrix $(N_q^T N_q)^{-1}$. These errors will tend to increase as hyperplanes are dropped from and added to the inverse, particularly when one or more of the hyperplanes in the inverse is close to being linearly dependent. As a result, the projected unit gradient vector s may have small components which do not lie in the desired intersection Q . As a result it may happen that after taking a step along s and adding a new hyperplane to Q the new point x^0 may violate some of the constraints which define the intersection Q .

The q hyperplanes H_i , $i=1, 2, \dots, q$, define the intersection Q and we suppose that the corresponding inverse matrix $(N_q^T N_q)^{-1}$ has already been computed as part of the method. The q values of $e_i(x^0)$, $i=1, \dots, q$, are computed.

These would all be zero except for the effect of rounding error, and therefore they will all be relatively small. Using $e(x^0)$ as defined a point x^1 is obtained from

$$x^1 = x^0 - N_q (N_q^T N_q)^{-1} e(x^0) , \text{ where}$$

$$e(x) = N_q^T x - b_q .$$

If there is no error in $(N_q^T N_q)^{-1}$ then it follows that $e_i(x^1) = 0$, $i=1, \dots, q$, and x^1 is the desired corrected point which satisfies all the constraints. It can be shown that if $|e(x^0)| \leq \delta$ then $|e(x^1)| \leq \delta^2$. This correction procedure is therefore essentially an error-squaring process. Since the inverse and $e(x^0)$ are computed as part of the regular method, the correction procedure, when needed, is obtained with less than $2n^2$ additional multiplications. Another approach to removing the effect of rounding error is to construct a new value for $(N_q^T N_q)^{-1}$ and P_q from $N_q \equiv [n_1, n_2, \dots, n_q]$ using the appropriate algorithm.

(iv) A useful comparison can be made between the efficiency of the gradient projection method applied to the linear programming problem and the elementary Simplex method. Let us define the linear programming problem as

$$\text{maximize } \left\{ p^T x \mid N^T x \geq b , x \geq 0 \right\} ;$$

$N : nxm ; x , p : nx1 , b : mx1 .$

Clearly, the linear objective function $f(x) = p^T x$ has a constant gradient given by $g(x) = p$. Furthermore, the step-size will always be determined by

$$\lambda_i^b = \min_i \left\{ e_i(x^k) / s_{n_i}^T \geq \delta > 0 \right\}$$

if $s_{n_1}^T > 0$ and $i = q + 1, \dots, m$.

At a vertex $x^0 \in R$ using gradient projection $q = m$ and the matrix N_m is square, so that $(N_m^T N_m)^{-1} N_m = N_m$. Furthermore, for any g , $P_m g = 0$. The necessary and sufficient conditions for a maximum thus reduce to

$$r = N_m^{-1} p \leq 0.$$

For x^0 not optimal in gradient projection, the vector n_ℓ (where r_ℓ is the maximum positive component of r) is dropped from N_m and the vector to be introduced is chosen from those n_i not in Q defined by λ_i^b . It can be shown that the gradient projection and the Simplex method are identical in the choice as to which vector is dropped from and which vector is added to the basis at each step. Therefore, the two methods will follow the identical vertex to vertex path for a linear problem started at a vertex.

For a vertex to vertex path in gradient projection the inverse matrix $(N_m^T N_m)^{-1}$ must be altered at each step. This is done by means of the recursion relations previously defined and requires approximately $4m^2$ multiplications per step. In Simplex only the matrix N_m^{-1} has to be altered, which requires approximately $3m^2$ multiplications per step. On this basis it is to be expected that Simplex will require less computing time per step but since the gradient projection method tends to cut across the interior of the convex region rather than always go from vertex to vertex the net result will be that although Simplex is somewhat faster for most linear problems, there will be certain types of linear

problems for which gradient projection will be faster, provided that we are not at a vertex.

CHAPTER VI

RESULTS AND CONCLUSIONS

The purpose of this thesis was to review and analyse critically the techniques available for the numerical solution of nonlinear systems, to optimize constrained nonlinear functions, and to develop practical methods for high-speed computers. In our approach to these problems we assume that a digital computer is available to perform the drudgery of computations invariably involved in any method of solution. This chapter is subdivided into three sections as follows:

(i) a discussion of the approach used in this thesis,

(ii) a condensed review of material covered and conclusions arrived at, and

(iii) an outline of the remaining problems in the field of nonlinear systems and nonlinear programming.

§ 6.1 The Approach Used in the Thesis

The two or several variable problem is, in general, a non-trivial extension of the single variable problem. An example of this is the great difficulty of generalizing Rolle's theorem to more than one variable.

Most natural processes depend upon several independent variables - contrary to the simplifying assumptions made by engineers and scientists - and, furthermore, these variables are generally not separable. Before the advent of modern automatic computing equipment the labour of computing with even a single independent variable was considerable and, with many variables, prohibitive. The very crudity of existing methods is adequate evidence of a groping and iterative approach to the numerical solution of these problems; the difficulties recounted in this thesis are by no means trivial and it seems unlikely that these problems will be completely

solved in the near future. Certainly an improvement of techniques should increase the range of problems which can be handled or significantly reduce the effort necessary.

Prohibitive labour is perhaps an adequate reason but another - and more significant - reason in the lag in the development of numerical methods of coping with multivariate functions is the relative unfamiliarity of the mathematics of several independent variables. Much of the existing mathematics applicable to problems in several variables is framed in highly abstract terms; hence, it is not immediately accessible to those numerical analysts who have entered the field from engineering or the physical sciences. This second reason has greatly influenced the approach followed in this thesis. We have devoted a major part of this thesis to topics not directly applicable to the nonlinear system and constrained nonlinear functions but these topics are required to form the necessary background.

§ 6.2 Review of Thesis and Conclusions

Before we attempted to solve the nonlinear system in several variables an attempt was made to prepare an adequate background in the algebraic problems. We mentioned the further difficulties that are encountered in the polynomial case because of ill-conditioning which could be adequately handled only by resorting to high-precision arithmetic. The iterative methods applicable to the linear system are included because of the importance of the extension of these methods to nonlinear problems.

The nonlinear system defined as

$$f(x) = 0 ; f(x) , x : nx1$$

is handled most effectively, in general, by the Conjugate Gradient Method. The other methods reviewed, such as Newton's, Steepest Descent, and Iterative methods can either not be applied because of stringent conditions on convergence or they converge very slowly in the region of a solution. The Conjugate Gradient method is applicable to computer programming and its property of quadratic convergence near a stationary value makes it the most suitable general purpose method available. The use of the projection matrix in solving the conjugation of successive vectors problem has been tested and found to be a practical approach to this problem. Experience in using this method suggests that ultimate computing efficiency is obtained through using low precision arithmetic in the early stages of the method and then increasing the precision as we approach a solution.

The more difficult problem of nonlinear programming was held in abeyance until the basic concepts and theorems in linear programming had been introduced. The only methods described for the linear programming problem defined as

$$\text{maximize } \left\{ p^T x \mid Ax \leq b , x \geq 0 \right\} ; A : mxn ;$$

$$x , p : nx1 , b : mx1 ,$$

were the Simplex and Dual Simplex methods although more powerful methods are now available. The notation applicable to the linear problem was extended to the nonlinear problem

as, e.g. the quadratic case:

$$\text{maximize } \left\{ p^T x - \frac{1}{2} x^T C x \mid Ax \leq b, x \geq 0 \right\} ;$$

$C : n \times n$ symmetric, non-negative definite, $A : m \times n$; $p, x : n \times 1$, $b : m \times 1$. The methods reviewed were Beale's extension of the Simplex method, the method of Feasible Directions, and the Conjugate Gradient method. The preferred method and the method programmed for the I.B.M. 1620 computer is the Conjugate Gradient Method.

§ 6.3 Remaining Problems

In Chapter III we attempted to pose the problem of solving the nonlinear system and to partially appraise the method of solution currently available. We noted that intuition and experience were invaluable assets in matching the most effective method to the particular problem. A mathematical result, urgently required, which would place the known numerical methods of solution on a firmer basis is a workable criteria for the existence and localization of roots of nonlinear systems in several variables. A further result which would be of immense benefit in the solution of the nonlinear system is a practical method for a "best" polynomial approximation to multivariate functions associated with e.g. a least squares fit.

In Chapter V our treatment of the nonlinear programming problem was restricted to the particular case of quadratic programming. Another equally important but more difficult case is convex programming - the problem of minimizing a convex function (or maximizing a concave

function) in a convex region. The methods available when the constraints are linear are the extension of Feasible Directions by Zoutendijk (1959) and Gradient Projection by Rosen (1961). An interesting approach to the convex programming problem is the Cutting-plane method by Kelley (1960). The basic idea behind this method is to construct local linearizations of the nonlinear constraints and then solve the simpler problem. The solution is then improved by more accurate linearizations to the constraints in the neighbourhood of the solution. The convexity assumption plays an important role in these problems; without such an assumption the existing methods either do not work or they lead at best to a local optimum. Kuhn and Tucker (1951) showed that the method of Lagrange Multipliers can be extended in such a way that a solution criterion can be given in terms of a saddle value problem and further developments using this approach is available in Arrow, Hurwicz, and Uzawa (1958).

BIBLIOGRAPHY

- Aitken, A. C., Proc. Roy. Soc. Edinb., 46, 289-305, (1926).
- *Arrow, K. J., Hurwicz, L., and Uzawa, H., Studies in Linear and Nonlinear Programming, Stanford Univ. Press, (1958).
- Beale, E. M. L., "On Optimizing a Convex Function Subject to Linear Inequalities", J. Roy. Stat. Soc., 17, 173-184, (1955).
- Bechenbach, E. F., (edit.) Modern Mathematics for the Engineer, University of California Extension Series, McGraw-Hill Book Co., Inc., Toronto : Ont. (1956).
- Bocher, M., Introduction to Higher Algebra, New York : The MacMillan Company, (1922).
- Brooker, R. A., "The Solution of Algebraic Equations on the Edsac", Proc. Camb. Phil. Soc., XLVII, 255-270, (1952).
- Burnside, W. S. and Panton, A. W., The Theory of Equations, Vol. I, Dublin and London : Dublin University Press, (Seventh Edition 1912).
- *Cheney, E. W. and Goldstein, A. A., Newton's Method of Convex Programming and Tchebycheff Approximation, Numer. Math., 1, 253-268, (1959).
- Curry, H. B., "The Method of Steepest Descent for Nonlinear Minimization Problems", Quart. Appl. Math., 2, 258-261, (1944).
- Dantzig, G. B., "Maximization of Linear Function of Variables Subject to Linear Inequalities", Chapter XXI in Koopmans (1951a), (1951b).
- _____, "Application of the Simplex Method to a Transportation Problem", Chapter XXIII in Koopmans (1951), (1951b).
- *Davidon, W. C., "Variable Metric Method for Minimization", Argonne National Lab. Rept. No. ANL-5990 Rev. Phys. Math. (TID-4500, 14th ed.), (1959).
- Dorn, W. S. "Duality in Quadratic Programming", Report NY08676 (Physics), New York University, (1958).
- Fadeeva, V. N., translated by Benster, C. D., Computational Methods of Linear Algebra, New York : Dover Publications, Inc. (1958).

- *Farkas, J., "Über die Theorie der einfachen Ungleichungen", Journal für die reine und angewandte Mathematik, 124, 1-24, (1902).
- Forsythe, G. E., "Solution of Linear Systems can be Interesting", Bull. Amer. Math. Soc., Vol. 59, 299-329, (1953).
- Fox, L., Husky, H. D., and Wilkinson, J. H., "Notes on the Solution of Algebraic Linear Simultaneous Equations", Quart. Jour. of Mech. and Appl. Math., Vol. 1, 149-173, (1948).
- Frost, P., An Elementary Treatise on Curve Tracing, London : MacMillan and Company, (1872).
- Gass, S. I., Linear Programming, McGraw-Hill, Toronto : Ont., (1958).
- Gill, S., "A Binary Form of Horner's Method", The Comp. Jour., Vol. 1, 84-86, (1958).
- Good, R. A., "Systems of Linear Relations", Soc. Ind. Appl. Math. Review, 1, 1-31, (1959).
- Haselgrove, C. B., "The Solution of Non-linear Equations and of Differential Equations with Two-point Boundary Conditions", The Comp. Jour., Vol. 4, 255-259, (1961).
- *Hestenes, M. R. and Stiefel, E., "Methods of Conjugate Gradients for Solving Linear Systems", J. Res. N. B. S., Vol. 49, (1952).
- Hildebrand, F. B., Methods of Applied Mathematics, Prentice-Hall Inc., Englewood Cliffs, N.J., (1952).
- *Hitchcock, F. R., "Distribution of a Product from Several Sources to Numerous Localities", Jour. of Math. Phy., Vol. 20, (1941).
- Kelley, J. E. Jr., "The Cutting-Plane Method for Solving Convex Programs", Soc. for Indust. and Appl. Math. Jour., Vol. 8, 703-712, (1960).
- *Koopmans, T. C. (edit.), Activity Analysis of Production and Allocation, Monograph No. 13 of the Cowles Commission, New York, N.Y., (1951).

Koopmans, T. C., "Optimum Utilization of the Transportation System", Econ. Vol.17, Supplement, (1949).

Kuhn, H. W. and Tucker, A. W., (edit.), Linear Inequalities and Related Systems, Ann. of Math. Studies No. 38, Princeton University Press, Princeton : New Jersey, (1956).

_____, "Nonlinear Programming",
Proceedings of the Second Berkely Symposium on Mathematical Statistics and Probablility (edit., J. Neyman), 481-492, (1951).

Lance, G. N., "Solution of Algebraic and Transcendental Equations on an Automatic Digital Computer", J. Assoc. Comp. Mach., 6(1), 97-101, (1959).

Martin, D. W. and Tee, G. J., "Iterative Methods for Linear Equations with Symmetric Positive Definite Matrix", The Comp. Jour., Vol. 4, 242-254, (1961).

Miller, G. A., "The Algebraic Equation", Monographs on Topics of Modern Mathematics, (edit. Young, J. W. A.), Dover Publications Inc., 211-262, 1955, (1911).

*Milne, W. E., Numerical Calculus, Princeton University Press, Princeton : New Jersey, (1949).

M. C. M., Modern Computing Methods, London : H.M.S.O., (1961).

Olver, F. W. J., "The Evaluation of Zeros of High Degree Polynomials", Phil, Trans. A, 244, 385-416, (1951).

Powell, M. J. D., "An Iterative Method for Finding Stationary Values of a Function of Several Variables", The Comp. Jour., Vol. 5, No. 2, 147-151, (1962).

*Riley, V. and Gass, G. I., Linear Programming and Associated Techniques, Johns Hopkins Univ. Press, Baltimore, (1958).

Rosen, J. B., "The Gradient Projection Method for Nonlinear Programming. Part I. Linear Constraints", J. Soc. Indust. and Appl. Math., 8, 181-217, (1960).

Rosenbrock, H. H., "An Automatic Method for finding the Greatest or Least Value of a Function", The Comp. Jour., Vol. 3, 175-184, (1960).

Spang, H. A., III, "A Review of Minimization Techniques for Nonlinear Functions", The SIAM Review, Vol. 4, No. 4, 343-365, (1962).

*Stigler, G. J., "The Cost of Subsistence", Jour. of Farm Econ., Vol. 27, (1945).

Temple, G., "The General Theory of Relaxation Methods Applied to Linear Systems", Proc. of the Roy. Soc. A, Vol. 169, 476-500, (1938).

Todd, J., (edit.), Survey of Numerical Analysis, McGraw-Hill Book Company, Inc., Toronto : Ont., (1962).

Turnbull, H. W., Theory of Equations, Interscience Publishers, Inc., New York, (1944).

Turner, L. R., "Solution of Nonlinear Systems", Ann. of the New York Acad. of Sc., Vol. 86, Art. 3, 817-827, (1960).

Vajda, S., Mathematical Programming, Addison-Wesley Publishing Company, Inc., Reading : Mass. and London : England, (1961).

_____, "Symposium on Linear Programming : An Outline of Linear Programming", J. Roy. Stat. Soc., 17, 165-173, (1955).

Ward, J. A., "The Downhill Method of Solving of $f(z)=0$ ", J. Assoc. Comp. Mach., 4(2), 148-150, (1957).

Whittaker, E. T. and Robinson, G., The Calculus of Observations, London : Blackie and Son, (1944).

Wilkinson, J. H., "The Evaluation of the Zeros of Ill-conditioned Polynomials", Numerische Math. I, 150-180, (1959).

*Wolfe, P., "Recent Developments in Nonlinear Programming", Rand Corp. Rept. No. R-401-PR, May 1962. To appear in : Proceedings of the 1962 Symposium on Mathematical Programming, McGraw-Hill, (1962).

Laguskín, V. L., Handbook of Numerical Methods for the Solution of Algebraic and Transcendental Equations, Pergammon Press : New York, (1961).

Loutendijk, G., Methods of Feasible Directions, Elsevier Publishing Company, Amsterdam, (1960).

_____, "Maximizing a Function in a Convex Region", J. Roy. Stat. Soc., 21, 338-355, (1959).

* These references were not available in the preparation of this thesis.

B29811